INTERACTIVE INTELLIGENCE®

·VONEXUS·

MICROSOFT®-BASED BUSINESS COMMUNICATIONS
An Interactive Intelligence Company

IC Reporting Data Dictionary

Technical Reference

Customer Interaction Center®
Enterprise Interaction Center®

Version 2.4

Last updated September 28, 2006

Abstract

This Technical Reference provides a concise and detailed description of the database tables IC uses to store and generate reports on historical data. This includes information stored for interactions, queue statistics, agent and user activity, line and line groups, administrative changes, and other Interaction Administrator configuration information. If you need to understand the structure and contents of IC tables, this is the document to read.

This document applies to one or more Interactive Intelligence and/or Vonexus products. Vonexus is a wholly owned subsidiary of Interactive Intelligence.

Copyright and Trademark Information

NOTICE

This product contains Information and/or data of Telcordia Technologies, Inc. (Telcordia) licensed to be included herein. Recipient acknowledges and agrees that (1) TELCORDIA AND ITS AFFILIATES MAKE NO REPRESENTATIONS, EXTEND NO WARRANTIES OF ANY KIND, EXPRESSED OR IMPLIED, AND ASSUME NO RESPONSIBILITY OR LIABILITY WHATSOEVER WITH RESPECT TO THE USE, SUFFICIENCY OR ACCURACY OF THE PRODUCT, (2) RECIPIENT SHALL MAKE NO CLAIM AGAINST TELCORDIA OR ANY OF ITS AFFILIATES WITH RESPECT TO THE PRODUCT, AND WAIVES ALL CLAIMS AGAINST TELCORDIA OR ANY OF ITS AFFILIATES WITH RESPECT TO THE PRODUCT, (3) IN NO EVENT SHALL TELCORDIA OR ANY OF ITS AFFILIATES BE LIABLE FOR ANY DAMAGES, INCLUDING ANY LOST PROFITS OR OTHER INCIDENTAL OR CONSEQUENTIAL DAMAGES RELATING TO THE PRODUCT, AND, (4) THIS AGREEMENT SHALL BE ENFORCEABLE BY TELCORDIA.

DISCLAIMER

INTERACTIVE INTELLIGENCE (INTERACTIVE) HAS NO RESPONSIBILITY UNDER WARRANTY, INDEMNIFICATION OR OTHERWISE, FOR MODIFICATION OR CUSTOMIZATION OF ANY INTERACTIVE SOFTWARE BY INTERACTIVE, CUSTOMER OR ANY THIRD PARTY EVEN IF SUCH CUSTOMIZATION AND/OR MODIFICATION IS DONE USING INTERACTIVE TOOLS, TRAINING OR METHODS DOCUMENTED BY INTERACTIVE.

Interactive Intelligence Inc.
7601 Interactive Way
Indianapolis, Indiana 46278
Telephone/Fax (317) 872-3000
www.ININ.com

Vonexus
7601 Interactive Way
Indianapolis, Indiana 46278
Telephone/Fax (888) 817-5904
www.vonexus.com

Interaction Center Platform Statement

This document describes Interaction Center (IC) features that may not be available in your IC product. Several products are based on the IC platform, and some features are disabled in some products.

Three products are based on the IC platform:

- Customer Interaction Center (CIC)
- Enterprise Interaction Center (EIC)
- Communité

While all of these products share a common feature set, this document is intended for use with all IC products, and some of the described features may not be available in your product.

How do I know if I have a documented feature?

Here are some indications that the documented feature is not available in your version:

- The menu, menu item, or button that accesses the feature appears grayed-out.
- One or more options or fields in a dialog box appear grayed-out.
- The feature is not selectable from a list of options.

If you have questions about feature availability, contact your vendor regarding the feature set available in your version of this product.

Table of Contents

Introduction

The Interaction Center (IC) Reporting Data Dictionary provides a concise and detailed description of the database tables IC uses to store and generate reports on historical data regarding interactions, queue statistics, agent and user activity, line and line groups, administrative changes, and other Interaction Administrator configuration information. In addition to documenting the structure and contents of IC tables, we try to help you understand the historical data IC collects. This document can also be helpful if you edit or maintain the shipped IC reports, or if you design your own reports.

In there are 19 tables and over 98 report variations of historical data collection in the reporting system. Understanding the details about each table and its relationship to other tables is the key to using the historical data for your reporting needs. The following list assists you in understanding the relationship of the collected data to the tables they are related to. The Type of data collected is described in great detail later in this document.

Interaction Center Database Tables	
Type of data collected	**Relationship to tables and logs**
Interaction Detail Data	Primarily included in the Call Detail table.
Agent State Data	Stored in the Agent Activity table.
Interval Queue Data	Used for the following tables: • Agent Queue Period Statistics Interval • Workgroup Queue Statistics Interval • Statistics Group Queue Interval
Interval Line and Line Group Data	Included in the Interval Line Statistic table and the Interval Line Group Statistics table.
Administrative Data	Included the Interaction Administrator Change Notification table and the IC Change Notification table.
Fax Data	Included in the Fax Envelope History table.
Configuration Mirror Data	Is included in the following tables: • User to Workgroup Relationship Mirror table • Account Code Mirror • Line Mirror • Line Group Mirror • Line Group to Lines Mirror.
IVR Interval and History Data	Includes reporting data for IVR menu navigation selections, date/time access, exit paths, time in menus and other statistics to help analyze IVR menu effectiveness.
Wrap Up Code Data	Data includes details about the amount of time agents spend in various wrap up states.

Interaction Center Database Tables	
Type of data collected	**Relationship to tables and logs**
Agent Queue Activation History	Includes details about agent activation and deactivation in each workgroup queue. This is independent of the agent state data.

About this document

Each section summary in this document describes the purpose of the tables it contains, along with an explanation of how IC generates records for each table. Specific information is included on the important statistics each table uses. This document lists columns in the order they appear in each table. Indexed columns are indicated in bold to make it easier for you to determine column importance. Indexes used by IC are listed separately.

Additional information about IC reporting is also available in the appendixes of this document.

Additional Statistics

IC does not store all the generated statistics in the historical reporting tables. Due to the large number of statistics available, IC only stores the most useful or relevant statistics in the tables. There are additional statistics available for custom applications that can be inserted into the custom columns of each table.

For example, the iStatsGroup table primarily shows ACD only statistics such as *nEnteredAcd*. The statistic for all entered interactions, ACD and non-ACD is *nEntered*. If you have a need for this statistic, you can insert it into one of the *iStatsGroup* custom columns and modify your reports to include it.

Contact Interactive Intelligence support for more information on using custom statistics, or see the *IC Advanced Reporting Guide* technical reference in the Technical Reference Documents section of the IC Documentation Library. The latest versions of technical reference documents can also be downloaded from our Support Web site.

Reports

Many customers request changes to the existing reports or the creation of new reports. Use the existing reports as a starting point to create custom reports. This is similar to using the existing default handlers as a starting point for your custom handler development. This document should help you in making reporting customizations, by providing an overview and description of the all the historical data generated by IC.

Although it is beyond the scope of this document, it is also important for you to understand the function, use, and limitations of the reports that ship with IC. For example, some of the reports within IC are intended for use that is more general, such as an enterprise running IC for its voice mail and PBX features. While these reports might be beneficial to a call center running IC, they would not be sufficient for presenting all the data needed to manage a call center. For more information about the existing reports, please refer to:

- The online reporting help available in Interaction Client on the Reports page. This is the end-user version of the Reporting help.

  Note Beginning with the Interaction Center 2.4 Client Feature Pack release, historical Crystal Reports are available in Interaction Supervisor. Interaction Center .NET Client and Interaction Client Outlook Edition users can now view, export, and print reports using Interaction Supervisor. If you are currently running the Interaction Client Win32 Edition, and upgrade to IC 2.4 Client Feature Pack release you can continue to view Reports in your Win32 client. However, if you upgrade to IC 2.4 Client Feature Pack and uninstall your Win32 client, you cannot reinstall the Win32 Client.
- To access the Reporting page in Interaction Supervisor, from the File menu click New. The New dialog, with the Reporting page, is displayed.
- The online reporting help available in Interaction Administrator on the Help pull-down menu. This is the administrator version of the reporting help.
- The *IC Reporting Guide.*
- The *IC Advanced Reporting Guide.*

Terms and Clarifications

Understanding Reporting Logs

The purpose of the reporting logs is to save historical data from the system. In Reporting logs are predefined entities that map to comma-separated value (CSV) files or SQL based database tables stored in either Microsoft SQL Server or Oracle. The references made in this document to SQL apply to MS SQL or Oracle. The references to tables also apply to CSV files, even though these files are not technically tables, the structure is the same.

After IC has successfully transferred the reporting logs or data to the tables, the system uses these tables as the source for the existing reports shipped by IC. You can customize these tables by using blank fields or custom columns available in several of the tables.

The predefined entities are named reporting *logs* rather than *tables* to clarify the process. Logs as a generic term to describe what purpose the data serves, and to isolate the system data from the method used to store the historical data.

Reporting logs should not be confused with the trace logs. Trace logs track the detailed actions of the different IC subsystems for debugging and other troubleshooting purposes. The information in the trace logs can contain reporting information, but mostly contains information on other IC subsystems. The trace logs are text-based and are not indexed or separated. The information is cryptic and unsuitable for reporting. Depending on the level of tracing that you require, the trace log files can grow to be quite large and the system is setup to routinely overwrite them each week. Normally, you should not need to be knowledgeable about trace log files because you typically only use them to troubleshoot system problems.

Understanding Interchangeable Terms

Other interchangeable terms used in this document include:

- Record or Row
- Column or Field
- Agent or User
- Period or Interval

Interactions

Interactions refer to calls, chats, e-mail, faxes, generic objects, various Web interactions, recorder interactions, and SMS interactions. In most cases, IC handles these interaction types similarly. However, in some cases the column name or even the table name, such as the Call Detail Table, still uses the term *call* when, in fact, multiple interaction types or values can be stored within the table.

For a list of Interaction Types and their mappings in the Call Detail Table, see "Calldetail Interactiontype column" onpage 8.

Database

The *Database* is considered the overall collection of tables and the relationship to other tables.

Log

Another place you will see the term *Log* used is in the case of the Call Log attribute associated with an interaction object. This text-based log contains an audit trail of actions performed on an interaction by the IC system, and is attached to the interaction object as an attribute only. It does not represent a table in any way. The Log Message tool step in a handler makes additions to the call log. This tool which is in the default IC system handlers, records and time-stamps certain events that happen to an interaction. When an interaction is disconnected, IC writes the call log attribute to a column, *CallEventLog*, in the Call Detail table.

Using the Switchover Process

When you configure two IC servers so that one is an automatic backup of the other using the switchover process, there should be no duplication of data or any problems relative to any report and configuration mirror data logging in IC. This is because the system that switchover considers the backup, operates in a suspended mode that suppresses the normal mirror and data logging activities.

Data Logging with Switchover

It is possible that duplicate data *might* be created, however, if you disconnect one of the two systems from the switchover hardware and boot it up as a stand-alone server. This stand-alone server then becomes a full and complete IC system that is identically configured to the other running server. This means that any operations that IC performs on this *separated backup* system will cause active mirror and logging activity on the shared database. This can result in the system inserting duplicate rows of data, or possibly, inserting wrong data into the tables. To make sure there is no conflict with data being generated by the two separated systems, when the IC systems are not in their switchover configuration, be sure to change the server Site ID.

Date/Time Columns

Date/Time columns might contain dates using the year 1970 in cases where no value was found or available. IC does not accept a null date/time, so there has to be a value. 1/5/1970 00:00:00 is the earliest date/time IC can represent, so IC uses it as an indicator of an unset date/time.

Interaction Detail Data

This section summarizes the purpose of the Interaction Detail Data and the Call Detail table. It explains how IC generates records for the table, and includes important statistics used by the table. It lists the columns in the order they appear in the table. Indexed columns are indicated in bold for better understanding of the columns'.

Data Source

The *data source* is a snapshot of interaction information at the time an interaction disconnects or ends. This data is collected one minute after disconnect. This delay allows any handlers that need to modify the attributes of the interaction *after* the disconnect and *before* the data is logged.

Type of information

The type of information includes end state, origination, duration, and last user active with an interaction.

Relationships to other tables and data

The relationship of the data in the Call Detail table to the data in all the other tables is very limited. Because IC only captures the last user's information, it cannot be related directly to data in any of the other tables except in **very** simple environments. IC does not reflect any transfers between distribution queues and agents in the Call Detail table or to any reports using this table.

The Call Detail table does have a relationship with the Account Code Mirror table, and supplies the Account Name in various reports.

Report Types using this data

Report types using this data include Call Detail, User Summary, DNIS, Account Code, Call, Call-Supervisor, and Supervisor Account Code. These categories of reports show simple interaction activity based on this data. These simple enterprise or summary activity reports will not be as complete as the interaction counts of Queue Period Statistics agent reports.

You should not apply the values stored in the Call Detail table to user activity values for purposes where stringent control is required, such as compensation. It might not represent a user's complete activity.

Call Detail Table

The Call Detail table contains data related to all physical interactions placed or received by the system. Each interaction creates one or more records in the table based on the number of objects it ultimately creates. An interaction creates a unique object, resulting in a new row, anytime one of the following actions occurs:

- The *MakeCall* function is used in a handler or by the IC Client.
- An inbound call is detected on a telephony interface card.
- A station is placed off hook.
- A conference call is created.
- An incoming chat is received.
- A station-to-station call creates two separate objects, one for each direction of the call, resulting in two records in the Call Detail table.

Most, but not all, of an interaction object's attributes, at the time the interaction is disconnected, are logged to the Call Detail Record table. Custom attributes are not recorded in this table, although it is possible for you to modify the system and place custom attribute information in the custom columns of this table.

Interaction Object Types

In addition to tracking information on calls, the Call Detail table contains information on chats, e-mail, faxes, generic objects, various Web interactions, recorder interactions, and SMS interactions.

Not all columns in the table are applicable to every supported object; some might not contain data for a specific interaction object type. However, supported interaction object types use the majority of the columns in the Call Detail table, but there are some fundamental differences between certain object types. The following differences occur for chat object types:

- All chats are inbound in direction. This means that an IC user/agent cannot initiate a chat.
- The *LocalNumber* field is not used for chats.
- Hold time for chats is not recorded to the Call Detail table, and as a result the information on hold time is defaulted to zero.

We have noted in the column when the information being stored in a column differs based on the type of interaction object.

Calldetail Interactiontype column

The Calldetail Interactiontype column, in this table, tracks the following codes for the Interaction type:

Calldetail Interactiontype Column	
Codes	**Description**
0	Call Object
1	Chat Object
4	Generic Object
5	E-mail Object
6	Callback Object
7	Instant Question
8	Browser Sync
9	Dual Form Web Interaction
10	Interaction Web Session
11	Monitor Object

How Remote Numbers and Names are Determined

Most remote number and name values in the Call Detail table are derived from some externally provided value such as the ANI, DNIS, and Caller ID, or by IC white pages lookup, custom handler changes, or a direct client edit.

How the remote number and name are provided depends on the type of service you use. T1 and E1 circuits-provided services are not standardized, and can be provider specific. That is why IC provides additional processing of the ANI/DNIS string in handlers. This allows installers to customize the handlers to do any additional processing that you might require for proper remote name and number processing.

Incoming Analog and ISDN Calls

For incoming calls on analog and ISDN lines, *RemoteNumberCallID* (in EIC as *Eic_RemoteTnRaw*) and *RemoteName* (in EIC as *Eic_RemoteNameRaw*) are obtained from the provider, when if available. If either is not available, IC will not set the corresponding attribute. If IC gets a remote number, it processes it further to generate *RemoteNumberFmt* and *RemoteNumber*.

The following are the Call Detail table remote name and number column name equivalent to the IC attribute, or call object attribute mapping IC uses to output the data analog and ISDN lines.

Call Detail Table Column Name	IC Attribute	Telephone Network Source
RemoteNumberCallID	Eic_RemoteTnRaw	Caller ID
RemoteName	TS Call object attribute RemoteName	Extended Caller ID
RemoteNumberFmt	Eic_RemoteTnDisplay	Display format of the Caller ID using the phone number configurations from IA.
RemoteNumber	Eic_RemoteTnNormalized	Normalized format of the Caller ID using the phone number configurations from IA.

Error codes

If caller ID is blocked or some other condition prevents delivery of the remote number, *RemoteNumberCallID* will contain one of the error code strings listed below and will leave the name, display, and normalized attributes blank.

Error code	**Meaning**
BLCKD	Caller id blocked
NOCLID	No caller id info available
OOA	Blocked, out of area
PRIV	Blocked, private number

In some cases, the provider might provide their own error messages in the Extended Caller ID information. These messages are passed on directly to the *RemoteName* column. TsServer also takes the remote number, if available, and does a white pages lookup and then appends that information, usually in the form of City, State but could also include the country, to the *RemoteName*.

Incoming T1/E1 Calls

If IC gets an ANI/DNIS string from the provider for an incoming call on an E1/T1 line, TsServer writes it to the *Eic_AniDnisString* attribute. The ANI/DNIS string attribute can be parsed using a pattern matching function in Interaction Administrator. This replaces the handler-based method used in prior versions of IC. IC still supports the handler-based method, but this newer functionality is easier and more convenient, and it is the preferred method. In the previous handler-based parsing method, the handler, *CustomParseT1AniDnisString.ihd*, provides two different example paths for parsing the ANI/DNIS string.

Regardless of the parsing method, IC uses the *Eic_AniDnisString* attribute string to set the following call attributes:

***Eic_AniDnisString* attribute string**	**Call attributes**
Eic_RemoteTnRaw	To set the ANI portion of the string.
Eic_RemoteNameRaw	Is also set to the ANI portion of the string.
Eic_RemoteTnNormalized	Set to a normalized format of the ANI portion of the string.
Eic_LocalTnRaw	Set to the DNIS portion of the string.
Eic_RemoteTnDisplay	Is set by processing the ANI value using the dial plan. If that value is blank, it is set by processing the ANI value through a white pages lookup. And if that value is blank, it is set to the ANI portion of the string.

If you use T1 or E1 circuits, you should verify that the ANI/DNIS string is being processed correctly, otherwise IC might be inserting incorrect or null remote name and number values in the Call Detail table.

The following are the Call Detail table remote name and number column name equivalent to the IC attribute, or call object attribute mapping IC uses to output the data when using T1 and E1 lines.

Call Detail Column Name	IC Atribute	Telephone Network Source
RemoteNumber	Eic_RemoteTnNormalized	Normalized format of the ANI
RemoteNumberFmt	Eic_RemoteTnDisplay	Display format of the ANI
RemoteNumberCallId	Eic_RemoteTnRaw	ANI
RemoteName	Eic_RemoteNameRaw (or TS Call object attribute RemoteName)	ANI
DNIS	Eic_LocalTnRaw	DNIS

SMDI

If you enable SMDI, on any line type, IC sets *RemoteNumberCallID* to the number returned. IC also uses this value to generate the values for *RemoteNumberFmt* and *RemoteNumber* columns. The system does not set *RemoteName*.

Outbound Calls (Interactions)

For outbound calls, IC sets *RemoteNumberCallID* to the dial string used to place the call. The system also further processes this number to generate *RemoteNumberFmt* and *RemoteNumber*. If IC passes a remote name value to TS with the **Make Call** request, the system sets *RemoteName* with this value. If system does not pass a remote name, IC performs a reverse white pages lookup using the remote number passed. If a *hit* is found, the system sets *RemoteName* to the returned value. Note that IC only performs the white pages lookup if no remote name is specified.

Physical Attributes

Log Identifier	Log Name	Table Name
10	CDR Log	CallDetail

Column Definitions

Column Name (Bold=Indexed)	IC Data Type	Output Data Type (Size)	Description
CallId	String	Char (18)	The ID key assigned by IC to this interaction record. CallIDKey includes the CallID embedded within it. That means you can search the Call Detail Record log for the original CallID number as seen in Interaction Client. The CallIDKey is comprised of the 10 digit CallID plus an eight digit date in the following format: YYYYMMDD. For example, a call placed on April 9, 2004 might show a Call ID of 2101990183 on Interaction Client. The resulting CallIDKey stored in the database would be that CallID plus the date: 2101990183**20040409**. Please see Appendix A for more information on how the call ID key is generated.
CallType	String	Varchar (20)	The originator of the interaction. The values include: *External* - The originator is external. *Internal* – The originator is local. *External Party* – The originator is external and the call is a

Column Name (Bold=Indexed)	IC Data Type	Output Data Type (Size)	Description
			conference call. *Intercom Party* – The originator is local and the call is a conference call. *Intercom* – The originator is local. *Unknown* – The originator type was undetermined. This usually means there was no originator. *ADSI* – The call was used to set a message waiting light on an ADSI phone.
CallDirection	String	Varchar (20)	Indicates if the interaction was *Inbound* or *Outbound* from the system. A value of *Unknown* indicates that the direction was not determined. Chats will all be *Inbound* in call direction. *Unknown* can also occur when using the station handset for voice mail or to place a call and no party-to-party call is actually dialed.
LineId	String	Varchar (50)	Identifies the line used for the interaction. The value will match the name of the line configured in Interaction Administrator. The value for chats is *Chat.*

Column Name (Bold=Indexed)	**IC Data Type**	**Output Data Type (Size)**	**Description**
StationId	String	Varchar (50)	Identifies the station used for the interaction. A value of *System* indicates a call that was not associated with a station. The *StationID* will be the value as configured in Interaction Administrator.
LocalUserId	String	Varchar (50)	The user identifier of the **last** user associated with the interaction. This value will only be set if there was a user logged in at the station placing a call, or the interaction was directed to an identifiable user. An empty field will occur if these conditions are not met. In the case of transfers or other handling that results in multiple users handling the interaction, the last person to handle the interaction gets associated with the interaction. For example, User1 talks to an outside party for 10 minutes and then transfers the call to User2. User2 talks for an additional 5 minutes and then hangs up. The call is associated with User2 for a total talk time of 15 minutes. This is because the call record is not intended to be an exact model of all the segments of a call. If the call is an intercom-to-intercom

Column Name (Bold=Indexed)	**IC Data Type**	**Output Data Type (Size)**	**Description**
			call, the local user is always the person who placed the call. When a call is placed from a station, the LocalUserId is set to the default user of that station even when the user is not logged into IC.
LocalNumber	String	Varchar (50)	Local phone number or extension number receiving or placing an interaction. For internal interactions within the IC system, the extension placing the call will be recorded as the *LocalNumber*. This field is blank for Chats.
LocalName	String	Varchar (50)	Local name of the station or user placing or receiving the call. This differs from *LocalUserId* because it can be assigned the station name if there is no user associated with the call. For internal calls within the IC system, the *LocalName* will be assigned to the person placing the call. See *LocalUserID* for more information.
RemoteNumber	String	Varchar (50)	External or extension number (or digits) dialed to reach a remote or internal party either for calls coming into the system or calls being placed out of the system. The format is the normalized version of the number best used

Column Name (Bold=Indexed)	IC Data Type	Output Data Type (Size)	Description
			for searching. This means that when searching for a number, the number to search for must contain the complete number of digits with NO punctuation (for example, country and/or area code, exchange, and number). For chats, the value will be the IP address of the initiating website. See the section "How Remote Numbers and Names are Determined" previously in this document for more information.
RemoteNumberCountry	Integer	SmallInt (2)	Number code field for the international code of the *RemoteNumber.* The value for this field is zero for chats.
RemoteNumberLoComp1	String	Varchar (10)	The first code field from the remote number. In the US, this will be mapped to the area code. In the UK, it will be the city code. Use will vary by nation and local. This field will be blank for chats.
RemoteNumberLoComp2	String	Varchar (10)	The second code field from the remote number. In the US, this will be mapped to the exchange. Its use elsewhere will depend on the standards in the local and the number maps created in IC. This field will be blank for chats.

Column Name (Bold=Indexed)	**IC Data Type**	**Output Data Type (Size)**	**Description**
RemoteNumberFmt	String	Varchar (50)	Formatted version of the *RemoteNumber*. The formatting is done using the IC number patterns defined in Interaction Administrator. See the section above on *How Remote Numbers and Names are Determined* for more information. For chats, the value will be the IP address of the initiating website.
RemoteNumberCallId	String	Varchar (50)	Caller Id number obtained from the telephone company. This is the value used to obtain the *RemoteName*. *NOCLID* is displayed if the caller id information cannot be obtained from the telephone company. *OOA* is displayed for out of area calls. Chats will display the IP address of the initiating website. See the section above on *How Remote Numbers and Names are Determined* for more information.
RemoteName	String	Varchar (50)	External or extension name dialed or entered. For intercom calls, the user name of the person dialed will be used (if the user is known.) If a station is dialed direct, only the station name will be used. Inbound calls that provide *Extended Caller Id* information will use the calling party name. *Unknown*

Column Name (Bold=Indexed)	**IC Data Type**	**Output Data Type (Size)**	**Description**
			Name will be displayed when *Extended Caller Id* is not available. An empty value is possible if the call was an *Intercom* call that was terminated before the remote party was specified. For chats, the name of the user, as supplied to the chat, will be displayed. This information is based solely on the users input. It is important to note that IC handlers or users can change the value of *RemoteName* and that it may not correlate directly to the *RemoteNumber*. See the section above on *How Remote Numbers and Names are Determined* for more information.
InitiatedDate	DateTime	DateTime (8)	Date/time that the interaction was originated. This is the time the interaction was created as an internal entity of IC. This time can be before there is an external connection to the call.
InitiatedDateTimeGmt	DateTime	DateTime (8)	Greenwich Mean Time-based initiated date and time. Should be used if calculations need to be able to ignore the effects of changes in time due to interactions spanning daylight savings time.
ConnectedDate	DateTime	DateTime	Date and time

Column Name (Bold=Indexed)	IC Data Type	Output Data Type (Size)	Description
		(8)	interaction connected to external network source or a system resource. This value is started when an interaction reaches a connected state within the IC system.
ConnectedDateTimeGmt	DateTime	DateTime (8)	Greenwich Mean Time-based connected date and time
TerminatedDate	DateTime	DateTime (8)	Date and time that the interaction terminated its connection to an external network source. This value is set on termination of the interaction, either from a local disconnect or a remote disconnect
TerminatedDateTimeGmt	DateTime	DateTime (8)	Greenwich Mean Time-based terminated date and time.
CallDurationSeconds	Integer	Int (4)	Interaction duration (connected time to terminated time) in seconds. This value is calculated using the GMT times, so it is not affected by changes in time due to daylight savings.
HoldDurationSeconds	Integer	Int (4)	Number of seconds that an interaction was in a HOLD state within the IC system. Chats will have a value of zero seconds
LineDurationSeconds	Integer	Int (4)	Total time that the line or station resource was in use (initiated time to terminated time). This includes ring time and other non-billable time on line.

Column Name (Bold=Indexed)	IC Data Type	Output Data Type (Size)	Description
DNIS	String	Varchar (50)	DNIS (Dialed Number Identification Service) number as obtained from the telephone network for any inbound call on DNIS enabled lines. See the section above on *How Remote Numbers and Names are Determined* for more information. This column will be blank for chats.
CallEventLog	String	Varchar (2000)	The call event log from the interaction object. This text-based log contains an audit trail of actions performed on the interaction by the IC system. Entries are made using the Log Message tool step in a handler. In the default IC system handlers, certain events that happen to an interaction are recorded and date/time stamped in the event log. Customizations may be made to add your own messages to the call event log although this column is limited to accepting a call event log that is no more than 2000 characters long (increased from 1024 characters in IC 1.3). Call notes manually entered in an open Interaction Client call window are not saved in this column. See the *CallNote* column for more information. The

Column Name (Bold=Indexed)	IC Data Type	Output Data Type (Size)	Description
			information for chats is generally the information entered by the user when they initiate the chat. This information includes name, address, email, and telephone number.
CustomNum1	Integer	Int (4)	Number reserved for customer customizations.
CustomNum2	Integer	Int (4)	Number reserved for customer customizations.
CustomNum3	Integer	Int (4)	Number reserved for customer customizations.
CustomString1	String	Varchar (50)	String reserved for customer customizations.
CustomString2	String	Varchar (50)	String reserved for customer customizations.
CustomString3	String	Varchar (50)	String reserved for customer customizations.
CustomDateTime1	DateTime	DateTime (8)	Date time value reserved for customer use. Null values are not allowed and may be represented by 1970 values.
CustomDateTimeGmt1	DateTime	DateTime (8)	Greenwich Mean Time-based date and time value reserved for customer customizations. Null values are not allowed and may be represented by 1970 values.
InteractionType	Integer	Int (4)	Code for the Interaction type for this

Column Name (Bold=Indexed)	IC Data Type	Output Data Type (Size)	Description
			record. 0 = Call Object, 1 = Chat Object, 4 = Generic Object, 5 = Email object, 6 = Callback object, 7 = Instant Question, 8 = Browser sync, 9 = Dual Form web interaction, 10 = Interaction Web session, 11 = Monitor object, 12 = Interaction Recorder object, 13 = SMS object.
SiteId	Integer	SmallInt (2)	Site identifier of the source for this record. Used in multi-site installations to distinguish the origin of the record. Configured when setting up IC.
SubSiteId	Integer	SmallInt (2)	Not used in IC 2.1. The purpose of this field is to divide use within a site, such as for multiple independent companies on one IC system. Until IC has such functionality, this field is zero and is reserved for future use.
I3TimeStampGMT	(System Supplied)	DateTime (8)	System supplied date and time when this record was sent to the IC Logging server for insertion into the database.
AccountCode	String	Varchar (50)	This is the account/billing code assigned to either inbound or outbound interactions. This code is user defined. There are options in Interaction Administrator for

Column Name (Bold=Indexed)	IC Data Type	Output Data Type (Size)	Description
			setting how account codes will be used.
PurposeCode	Integer	Int (4)	This column is used to track the purpose of the call. This is a system-defined code. It is currently only used for fax objects but will be expanded to include calls made on tie lines.
DispositionCode	Integer	Int (4)	This code is used track how an interaction ended in the system. This is a system-defined code.
CallNote	String	Varchar (1024)	This field contains any note manually entered for a call in the Interaction Client call notes window. Notes entered that are greater than 1024 characters will be truncated. The information in this field is currently not being used by any default reports in the system.
WrapUpCode	Integer	Int (4)	This contains the code entered by an agent during or after an interaction to indicate the purpose of the interaction. If multiple agents handle the call and enter a wrap-up code, only the last agent’s entry is stored in the table. If wrap up was required for the call, and no value was entered the value logged is the special code “NS” to represent not specified. A blank, or NULL, or "-"

Column Name (Bold=Indexed)	IC Data Type	Output Data Type (Size)	Description
			indicates that no Wrap-Up code was either required or specified.

Index Definitions

Index Name	Index Type	Column Name	Order
ConnDate_IDX	clustered non-unique	ConnectedDateDate	Ascending
		SiteID	Ascending
		SubSiteID	Ascending
LocalUserID_IDX	non-clustered non-unique	LocalUserID	Ascending
I3TimeStampGMT_IDX	non-clustered non-unique	LocalUserID	Ascending
I3TimeStampGMT_IDX	non-clustered non-unique	I3TimestampGMT	Ascending
		SiteID	Ascending
		SubSiteID	Ascending

Agent State Data

IC records Agent state changes in the Agent Activity table. The terms *State* and *Status* are not the same in the context of the agent state data. State changes usually correspond to status changes, but it is possible for an agent's state to change, and a new record created, even when the agent's status does not. For example, Interaction Client login/out events can change the agent state without changing the agent status, provided the status they are in at the time of the login/out event is defined as persistent. If a user logs out of IC without being in a persistent status, the system automatically sets the user's status to the last persistent status message used.

The Agent Activity table tracks changes in agent state based on changes made in:

- **Agent login / logout**—usually by starting or ending the Client but can also be done manually over the phone.
- **Agent status**—such as After call work, Do Not Disturb, and ACD-Agent not answering. Also known as the *StatusKey* in the table.
- **ACD login / logout**—when an ACD agent status is selected/unselected.

A new record is generated anytime one of the above conditions change. Technically, the current state is defined as the unique set of *StatusKey, LoggedIn, AcdLoggedIn, StatusDnd,* or *StatusAcw* values. If any of these values change it causes an ending of the current state and the start of a new state. Each change of state signals a new state individually for that agent.

Data in the Agent Activity table is not interval-based because there is no set duration for each state. States of zero duration will also be generated by some events such as logging into IC and logging out of IC in order to help separate these events from other agent state events. For interval- and queue-based agent statistics, see the *tAgent* statistics in the interval queue data tables.

Agent Activation is Outside of Agent State

Agents can optionally be activated and deactivated individually in each workgroup queue independent of the agents' status or agents' state. This feature enables authorized agents, supervisors and administrators to manually activate or deactivate agents from specific ACD workgroup queues, regardless of the agent's ACD login status or user status. For more information about agent activation, see the section on "Agent Queue Activation History" later in this document.

Data Source

The data source is delayed and event driven by change of user state. The previous state is recorded any time a new state is selected or automatically applied.

Type of information

The Type of information includes login indicator, status used, time starting and ending the status, along with the calculated duration in seconds. There are also flags that indicate if the status was an ACD, ACW, or DND status.

Report Types using this data

Agent and Agent Supervisor reports show this information in a detail and summary form.

Agent Statuses

IC defines agent statuses as the list of user status messages that you can configure in Interaction Administrator. By default these are:

- ACDAgentNotAnswering
- At a Training Session
- At Lunch
- At Play
- Available
- Available, Follow-Me
- Available, Forward
- Available, No ACD
- Away from desk
- Do Not Disturb
- Follow Up
- Gone Home
- In a Meeting
- On Vacation
- Out of the Office
- Out of Town
- Working At Home

It is possible to add additional custom statuses. The attributes associated with a custom status will determine how the system tracks it in the Agent Activity table.

Any status with the *Status allows ACD calls* box checked affect the ACD login state. The Available status is the only *default* agent status that affects ACD login/logout. If you select or deselect Available, it affects the agents ACD login state. This is separate from the Interaction Client login state and the agent activation state.

IC also shows agent status changes when monitoring agents that are not shown in the Agent Activity table, as they are not actual changes of user state. These are usually shown as a change in the icon next to the person's name being monitored. Examples include: *On the Phone, Agent Not Logged In, Agent on ACD Call*, and *Agent on OtherACD Call.*

Note that the current state of the agent is not recorded until the user or system time in that state ends. However, time in a state does not incur indefinitely. The individual states of all the agents defined on a system are written out once a day at midnight to avoid having indefinitely long periods associated with a state. 24 hours should be the longest time recorded for a state

Agent Activity Table

Most often, IC adds a new row to this table when an agent state changes. However, it is possible for some of the values that determine the state to change rapidly. In order to preserve the uniqueness of each row, IC uses a sequence number, *SeqNo*. This *SeqNo* preserves the state order when multiple state changes for an agent occur in the same second. This value starts at zero and this is the value that you will see in most cases. If there is more than one state change to be reported in a second, the system increments the value by one for each additional state change reported in that second. This means that records are unique by *Site*, *SubSite*, *UserId*, *DateTimeGMT*, and *SeqNo*. Nothing else defines the uniqueness of the row. Changes in other values might cause IC to output a new row, but only these values define unique rows in the database.

Physical Attributes

Log Identifier	**Log Name**	**Table Name**
80	Agent Activity Log	AgentActivityLog

Column Definitions

Column Name (Bold=Indexed)	**IC Data Type**	**Output Data Type (Size)**	**Description**
UserId	String	Varchar (50)	User identifier of the agent whose state has changed. This is the User ID as defined in Interaction Administrator.
StatusDateTime	DateTime	DateTime (8)	The date/time the agent state started this current state. The current state is defined as the unique set of StatusKey, LoggedIn, AcdLoggedIn, StatusDnd, or StatusAcw. A change of any of these values will cause an ending of a state and the start of a new state. More accurately, this is "state" Date Time.
StatusDateTimeGMT	DateTime	DateTime (8)	StatusDateTime, as previously mentioned, adjusted to Greenwich Mean Time. More accurately, this is "state" Date Time GMT.
ChangedStatus	Integer	SmallInt (2)	Flag indicating that the StatusKey changed in this state as compared to the previous state. 0 = No change, 1 = Changed.
ChangedStatusGroup	Integer	SmallInt (2)	Flag indicating that the StatusGroup changed in this state as compared to the previous state. 0 = No change, 1 = Changed.

Column Name (Bold=Indexed)	IC Data Type	Output Data Type (Size)	Description
ChangedLoggedIn	Integer	SmallInt (2)	Flag indicating that the agent's IC logged in state changed in this state as compared to the previous state. 0 = No change, 1 = Changed.
ChangedAcdLoggedIn	Integer	SmallInt (2)	Flag indicating that the Agent's ACD logged in state changed in this state as compared to the previous state. 0 = No change, 1 = Changed.
StatusKey	String	Varchar (50)	The status name associated with this state. Agent statuses are defined as the list of user status messages configured in Interaction Administrator. This database value is not localized, because it would be wrong if a customer were using more than one locale on a server. That is why it is a KEY. The value is localized on the report using the multi-locale support for status messages.
StatusGroup	String	Varchar (50)	The status group of the StatusKey. A status group is any grouping of agent status messages. A status message can only belong to one status group so this value relates directly back to StatusKey. There are five predefined status groups in IC: Available, Break, Followup, Training and Unavailable. Beyond just grouping statuses, status groups provide a way to track specific time in a status as part of the Queue Period statistics information of an agent or distribution queue.

Column Name (Bold=Indexed)	**IC Data Type**	**Output Data Type (Size)**	**Description**
			See the tStatusGroup columns in the IAgentQueueStats and IWrkgrpQueueStats tables for more information. It is possible to define your own custom groups. See the online help in Interaction Administrator for more information. It is also possible to make the group to status mapping a one to one mapping. No grouping is actually forced, just encouraged.
LoggedIn	Integer	SmallInt (2)	Flag indicating that the agent was logged in to IC for this state. 0 = False, 1 = True.
AcdLoggedIn	Integer	SmallInt (2)	Flag indicating that the StatusKey of this state allowed the agent to take ACD interactions. 0 = False, 1 = True.
StatusDnd	Integer	SmallInt (2)	Flag indicating that the StatusKey of this state was a Do Not Disturb status. 0 = False, 1 = True.
StatusAcw	Integer	SmallInt (2)	Flag indicating that the StatusKey of this state was an After Call Work status. 0 = False, 1 = True.
EndDateTime	DateTime	DateTime (8)	The end date/time of this state. The end of the state is signaled by some change of state that would change one of the values recorded in this log. That could be a change of StatusKey, LoggedIn, AcdLoggedIn, StatusDND or StatusAcw. It is possible that more than one of these was changed, or that only one of these has changed.
EndDateTimeGMT	DateTime	DateTime (8)	EndDateTime (see above)

Column Name (Bold=Indexed)	**IC Data Type**	**Output Data Type (Size)**	**Description**
			adjusted to Greenwich Mean Time.
StateDuration	Integer	Int (4)	Duration, in seconds, the previous state was active. The agent's current state duration will not be logged until their state has been changed.
SeqNo	Integer	SmallInt (2)	Sequencing number to preserve the state order when state changes for a user occurred in the same second. This value starts at zero and zero is the value that will be seen in most cases. If there is more than one state change to be reported in a second, the value will be incremented by one for each additional state change reported in that second.
I3TimeStampGMT	(System Supplied)	DateTime (8)	System supplied date and time when record was sent to the IC Logging server for insertion into the database.
SiteId	Integer	SmallInt (2)	Site identifier of the source for this record. Used in multi-site installations to distinguish the origin of the record. Configured when setting up IC.
SubSiteId	Integer	SmallInt (2)	Currently not used. The purpose of this field is to divide use within a site, such as for multiple independent companies on one IC system. Until IC has such functionality, this field is zero and is reserved for future use

Index Definitions

Index Name	Index Type	Column Name	Order
I3TimeStampGMT	Clustered non-unique	I3TimeStampGMT	Ascending
		SiteId	Ascending
		SubSiteId	Ascending
IEAgentActivityLog	Non-clustered	UserId	Ascending
		SiteId	Ascending
		SubSiteId	Ascending

Interval Queue Data

This category of data includes the IAgentQueueStats, IWrkgrpQueueStats and IStatsGroup tables. These tables contain interval-based summary records that represent the activity that occurred for queues or queue-like objects in IC. Distribution queues, custom statistics groups (or STIDs), and agents/users all represent types of queues in IC. The values IC collects for these queues include standard ACD management values such as:

- Counts of interactions that entered the queue
- Answered and abandoned counts
- Interactions that flow out
- Interactions that span the intervals

IC also tracks most events that trigger counts regarding the time until the event occurred: tracking the total time to answer, abandon, or flow out.

Other examples include:

- Time spent on interactions.
- The time interactions spent on hold at an agent.
- Time spent in an ACW (or A*fter Contact Work*) state associated with an interaction.

Data Source

The system writes Interval data for agents, distribution queues, and statistics groups to the tables using automated logging functions within StatServer. The output values can be customized in Interaction Administrator.

Type of information

The type of Interval Queue Data information includes counts of interaction entry, answer, abandon, or flow out, and it includes counts of the *time taken* to answer, abandon, or flow out. Time on interactions, time interactions spent on hold with an agent, and time agents spent in after contact work (ACW) are included, as well. In the case of agent and distribution queues, it includes the staffing times for agents and time the agents were in particular states relative to the interval.

Report Types using this data

Agent performance queue and DNIS categories use Interval Queue data. This includes all of the Queue Period Statistics reports.

Intervals

The Interval Queue Data tables are interval-based, meaning that, rather than writing a record for every event, IC stores totals for the events in memory for a period of time known as the interval. At the end of the interval, the system writes a summary record for all the events that occurred during the interval for an interaction, to the tables. We often refer to this interval as the Queue Period

Statistics (QPS) interval. For more information on setting the interval and other details of the Queue Period Statistics process, see Appendix B.

Comparing Queue Data

When you are trying to compare the data stored in the Interval Queue Data tables, it is easy to be confused by the relationships that the various types of queues have to each other, and how this affects the data stored in the tables. In addition, the meaning of many of the similarly named columns might be different relative to the level of queue assignment that each table represents.

Interval Queue Data Tables	
Data Table	**Purpose**
IAgentQueueStats	Stores information gathered for agent/user queues.
IWrkgrpQueueStats	Stores information gathered for distribution queues.
IStatsGroup	Stores information gathered for statistics groups.

Model of hierarchy

A simple model of the queue hierarchy is:

> **Statistics Groups**, which can contain **distribution queues**, which can contain **agent/user queues**.

In particular, it is important for you to understand the relationship that an agent/user queue has with any distribution queue of which the agent is a member. Agent/user queues are on a different queue level from distribution queues. When the system assigns interactions to a distribution queue and then goes through the ACD agent assignment process, the system also assigns them to agent/user queues in order to be answered. In this regard, you can say that agent/user queues are a sub-level of distribution queues. Although it appears that IC assigns the interaction to only one queue, the distribution queue, it is actually assigning the interaction to two queues, with the addition of the agent/user queue assignment, before being answered. The statistics in the tables show the affect of these assignments. For instance, you can only count an interaction as answered once relative to each queue assignment. This means that if an agent takes a distribution queue interaction that has already been answered once by another agent in the distribution queue, IC re-assigns that interaction to the other agent/user queue. Because the interaction has not left the distribution queue, IC does not re-assign it at the distribution queue level. The interaction maintains its current assignment at the distribution queue level, and even though the interaction is answered again by a different agent, it is not answered again by the distribution queue.

An example of a common mistake arising out of not understanding these queue level assignments happens when one compares the *nAnsweredACD* count of a distribution queue with the sum of the *nAnsweredACD* counts of all the agents who are members of the distribution queues. You can only do this by comparing the IwrkgrpQueueStats table-based Distribution Queue Performance reports with the IagentQueueStats table-based Agent Performance reports. Customers often report that they see a discrepancy in the total number of interactions answered by the distribution queue, because the total number of answered interactions in the Distribution Queue Performance reports often appears higher than the total seen in the Agent Performance reports. However, the discrepancy is due to the difference in the type of queue each report is reporting. Distribution Queue Performance reports are based on interactions handled by distribution queues. Agent Performance reports are based on interactions handled by Agent/User queues.

When an agent first answers a distribution queue interaction, IC counts the interaction towards the distribution queue's answered interaction total AND the agent's queue answered interaction total. However, if the agent then transfers the interaction to another agent, who is also a member of the same queue, IC does not count the interaction again towards the distribution queue's answered interaction total, but it does count it towards the second agent's answered interaction total. Same interaction, two different agents, equals one interaction for the distribution queue and one interaction for each agent queue. If you try to add up the total of all interactions answered by agent queues, hopefully you can see that this total might not match the total number of interactions answered by the distribution queue.

The only way you can get the same interaction to be counted as answered multiple times for a distribution queue would be to transfer the interaction out of the queue in a way that allows it to be reassigned to the distribution queue. Another way of saying this is that IC only counts an interaction as answered once relative to a queue assignment. This rule applies to all types of queues: distribution queues, agent/user queues, and the special statistic identifier queues.

Report Groups

Report groups provide a way to sub-group interactions associated with a queue. You can use them to separate interactions based on, for example, DNIS or skills. A report group is simply an attribute of an interaction that IC assigns to the interaction before placing it in a queue, either by the system automatically or by a handler using the Assign Report Group tool. You cannot change the report group of an interaction while it is in a queue. In the Interval Queue Data tables, agent queues, based on agent membership, automatically generate one report group row per interval for each distribution queue. Distribution queues generate a row for each custom report group assigned to the interaction entering the queue. A special "*" or summary report group represents the sum of all activity for the agent or distribution queue in the interval based on the cType hierarchy implemented. There will be a "*" row for every level of hierarchy. See Record Type Hierarchy later in this document for more information.

Record Type Hierarchy

The Interval Queue Data tables (*IAgentQueueStats*, *IStatsGroup*, and *IWrkgrpQueueStats*) contain two columns, *cHKey3,* and *cHKey4*, to expand the number of possible record types that the system stores in these tables. This record type hierarchy structure allows for reporting by up to four levels of information. An example is, a report that groups interactions by agent, distribution queue, interaction type, and wrap up code.

The *cName, cReportGroup*, *cHKey3*, and *cHKey4* columns form a flexible four-part key that uniquely defines each row within the record type hierarchy. The elements in the key form a hierarchy of record type values from the most significant to the least significant identifier. Although this hierarchy key uniquely defines the row for the purposes of the hierarchy, a unique indexed row in the table requires the additions of the *SiteId*, *SubSiteId*, *dIntervalStart* and *I3TimeStampGMT* values.

cType remains as the type of queue (agent/user, distribution queue, and statistics group) and defines the overall hierarchy. For this reason, we often refer to the hierarchy as the cType hierarchy. *cName* is the first level of the cType hierarchy and the system uses it to store the name of the agent, distribution queue, or statistics group associated with the record. *cReportGroup* is the second level of the cType hierarchy and the type of value the system stores changes, depending on the table. *cHKey3* is the third member and *cHKey4* is the fourth member of the key and the type of value the system stores in these columns also changes, depending on the table. For information on how you can use the cType hierarchy for each queue type, see the examples under each Interval Queue Data table description.

The *cReportGroup* column uses a single asterisk (*) to represent all interactions for the *cName*, providing a summary group. This is consistent with other tables, such as ILineGroupStats, and it enables easier localization.

There is a "-", or unspecified output row that IC usually uses for tracking interactions that do not have any report group assigned. These unspecified "-" rows only appear when you must log a child in the hierarchy, but there is no specific value for the parent. An example is logging an interaction type without setting any report group. The keys would have to be “Distribution queue”, "-", “Interaction Object”, and similar values.

By default, IC does not log interaction types in logs 90, 91 and 92 unless you specify the system to do so. Interaction types are logged automatically in log 10. To activate the logging of data by the interaction type, from **Interaction Administrator** open the **Server Configuration** dialog. On the **Report Configuration** page, select the **Turn on Media Type Reporting** check box.

When you activate the data collection, the hierarchies are as follows:

- A = UserId / Distribution Queue / Report Group / Interaction Type
- W = Distribution Queue / Report Group / Interaction Type
- S = STID / Report Group / Interaction Type

When you do not activate the data collection, the hierarchies are as follows:

- A = UserId / Distribution Queue / Report Group
- W = Distribution Queue / Report Group
- S = STID / Report Group

Interaction Types

Interaction types include the following possible values:

0 = Calls

1 = Web chats

4 = Generic object

5 = Email object

6 = Callback object

7 = Instant Question

8 = Browser sync

9 = Dual Form web interaction

10 = Interaction Web session

11 = Monitor object

12 = Interaction Recorder object

13 = SMS object

How an Interaction becomes an ACD Interaction

You should note that the Interval Queue Data tables include columns of ACD and non-ACD data, and sometimes the difference is subtle or difficult for you to determine. What distinguishes an ACD interaction from a non-ACD interaction is whether the ACD tools in a handler acts on the interaction. For example, an interaction that the system transfers directly to an agent rather than to a queue would be a non-ACD interaction. Interactions that IC do not distribute via the ACD tools can still appear on ACD distribution queues, so it is possible for you to have both ACD and non-ACD values relative to these queues. For the most part, IC stores data for interactions that are explicitly ACD interactions in columns. The names of these columns in the tables include *ACD* as a suffix, or as a portion of the name. IC places data for non-ACD related interactions under either the *ALL* group row totals or stores the data in columns in the tables where the name does NOT include *ACD* as a suffix, or as a portion of the name. It is possible, however, for IC to store ACD related data for interactions in columns that do not include the *ACD* suffix, or *ACD* as a portion of the name.

Here is a comparison between two columns that appear to be the same, *nAnswered* and *nAnsweredAcd.* The column *nAnswered* includes ACD and non-ACD interactions that were answered, whereas *nAnsweredACD* includes only ACD interactions that were answered.

The columns *tAgentLoggedIn* and *tAgentAvailable* are examples of columns that do not include *ACD* in their name because the columns relate specifically to the agent and not the interaction type.

The column *tAcw* is an example of a value that is typically related to ACD interactions because it is time accumulated in an *After Contact Work* state that usually happens after an ACD interaction, but it does not have to be after an ACD interaction. Because of this ambiguity, we do not include *ACD* in its name.

tAgent Values

IC collects Agent staffing or availability values, *relative to the queue and interval*, also in the IAgentQueueStats and IWrkgrpQueueStats tables. These values can include:

- Time logged in to the queue during the interval.
- Time available during the interval to take an ACD interaction.
- Time spent in various other states or statuses associated with the queue by interval.

We consider most of the *tAgent* columns to be agent-staffing statistics because you can use them to track or forecast—usually using third-party workforce scheduling software—agent staffing levels. The agent’s state, which is the sum of all ACD activity, interaction activity, and logged-in state rolled up together, is the only thing that controls these numbers. The interactions agents place or receive help determine the agent state, but these interactions do not directly determine the values in the *tAgent* statistics. Rather it is the ACD system and other parts of the system that determine if the agent is available. Note that the tAgent values cannot be compared to the information IC stores in the on Agent Activity Table, except in very simple environments.

For non-interval-based agent state data, see the section on "Agent Activity Table," earlier in this document.

Delayed Interactions that Span an Interval

Because the Interval Queue Data tables consist of records that show the sum of interaction activity for an interval, you might wonder what happens to interactions that span the interval. An example of a spanned interaction, also known as a delayed interaction, is an interaction that enters the ACD queue in the first interval, but is not answered until the second interval. This interaction shows up in the *nEnteredAcd* column of interval 1, but does not show up in the *nAnsweredAcd* column of interval 1. However, in interval 2 it registers in *nAnsweredAcd,* but not *nEnteredAcd*.

Since *nEnteredAcd* includes interactions where the system has not yet determined the end condition (any of the delayed interactions might be answered or might be abandon in the next interval), you need to count Total Answered (*nAnsweredAcd*) + Total Abandon (*nAbandonedAcd*) + Total Overflow (*nFlowOutAcd*) to get a count of the total number of interactions handled in an interval. This formula is better than *nEnteredAcd* alone because it removes the ambiguity of the delayed interactions.

The effect of delayed interactions is less severe in summary reports that include multiple intervals, such as a summary report for an entire day, unless your site handles interactions 24 hours a day.

Abandons

Abandoned queue interactions occur on a distribution queue when a queue interaction disconnects before it enters a *Client_Connected* state (that is, an agent or user picks it up). In the Queue Period Statistics reports, abandons are the number of ACD related queue interactions that abandoned the queue during the interval.

Abandons include:

- Disconnects before the queue interaction is connected to a user or agent. This only includes remote disconnects. IC has special processing to count local disconnects as answered, then disconnected.
- Interactions that go to a Voice mail box assigned to the ACD. To the ACD system, they are simply seen as queue interactions that disconnect while on the workgroup queue. To avoid having voice mail interactions count as abandons you must first transfer the interaction out of the queue. This is usually to a specially-setup, voice-mail-only queue. Voice mail interactions will then show up as flow outs rather than abandons.
- Interactions that go to a voice mail box assigned to the user/agent queue where the agent never answers the interaction.
- Interactions in the queue that might—or might not—be waiting for an agent that eventually remote disconnect without ever speaking to an agent. An example would be a caller listening to an IVR, and hangs up.
- Queue interactions that transfer from one workgroup queue to another workgroup queue, where no agent ever answers the queue interaction in the first workgroup queue, will not show up as abandons. They will only show as flow outs.

Flow Outs

IC defines flow outs as queue interactions that are removed from a queue without reaching an inactive state. Queue interactions have an inactive state that is used to mark the interaction for deletion by the ACD subsystem after the interaction has either been answered or abandoned. The most common reason for an interaction to leave the queue and not reach an inactive state—a flow out—is a queue interaction that is transferred. A transferred interaction is an interaction that is moved to another distribution queue or user queue by anything other than the ACD subsystem. This includes queue interactions that transfer from one distribution queue to another distribution queue, whether or not an agent answers the queue interaction in the first distribution queue. A user can transfer an interaction manually, or a system handler can transfer an interaction automatically.

Other reasons for flow outs:

- IC considers interactions that are grabbed from the queue to be flow outs. A grab is a flow out that was caused by a non-owner of the interaction manually taking ownership of it. This can be another member of the distribution queue manually taking the interaction or any IC user with sufficient rights to take ownership of the interaction. Note that ACD interactions waiting for an agent are not owned by any agent until the ACD subsystem assigns the interaction to an agent.
- When looking at agent queue data, an agent that did not answer a queue interaction (which then went to the next available agent) will show the queue interaction as a flow out on their user queue. This type of interaction also counts towards the *nNotAnsweredAcd* statistic on their user queue.
- Another possible reason for a flow out to occur is from hunt group alerts. A multiple alert for hunt group places the same interaction on multiple user/agent queues. IC considers the interaction a flow out in each of the user queues where the users do not answer. This means a large number of flow outs can be generated for each member of the hunt group when using this type of interaction routing. Note that these flow outs would not appear on the distribution queue statistics of the queue that is setup as a hunt group.

Interactions that IC does not considered flow outs:

- ACD interactions that go to voice mail, unless they are transferred to another queue first.
- An abandon interaction is never counted as a flow out, because when a queue interaction abandons, the system marks it inactive and removes it from the queue.
- Queue interactions that span one of the Queue Period Statistics intervals are not removed from the queue, so IC does not consider them flow outs. You can determine how many interactions spanned a given interval by looking at the combination of *WaitAlertAcdQueue*, *ActiveAcdQueue,* and *HeldAcdQueue* interactions, with the start or end prefix added so you can look at it from either the previous or current interval.

Grabbed and Transferred Interaction Counts

The value for flow outs on ACD queues is primarily made up of transferred and grabbed interactions. However, the individual statistics for grabbed interactions and transferred interactions are not currently very exact. IC still stores values in the database for these events but you should not consider them reliable. The statistic for flow outs is reliable, however you cannot break it down to show how many flow out interactions are the result of grabs or transfers.

There are various ways you can avoid grabbing interactions. You can add the users that are grabbing the interactions to the queue, and then assign skills to the interaction and the users so these new users are the last ones the system assigns the interactions. By doing this, you avoid the problem of tracking interactions that are grabbed, by reducing or eliminating grabs.

Conference Calls and Transferred Interactions

IC currently does not track the number of interactions that are transferred within the distribution queue. IC considers an ACD interaction transferred between agents within the same group as counting towards the agent's answered total, but does not count these transfers towards the overall ACD queue total for answered interactions.

This is not because IC does not know the source. IC does this by design because it is not correct to count the call as entering the distribution queue more than once, when it clearly does not. The call never leaves the distribution queue when it is transferred between agents, and as far as a manager is concerned, the distribution queue has only received one call. Since the distribution queue only received one call, IC does not count multiple answers of the call or the counts will not balance.

IC never again counts an interaction as answered within the ACD distribution queue, even if it is transferred to another agent, once an interaction reaches a *Client_Connected* state in that ACD distribution queue. An interaction transferred directly to an agent rather than to a queue is a non-ACD interaction.

For conference calls, the system treats the original call as an ACD interaction, but the conference call, unless an agent makes it directly to an ACD queue, is a direct call to an extension or an external number and is treated as such. If another agent in the queue joins the conference, IC considers it a non-ACD interaction for the agent. A call is only an ACD interaction if it ACD tools in a handler have manipulated it. If the original agent drops out, it does not change the call status. The call remains an ACD interaction. The original agent who answered the call gets all of the ACD statistics related to the call. The call is a non-ACD interaction for everyone else on the conference call.

Call Direction Counters

IC counts consult transfers as *nInternToInternCalls* for the agent that places the consult interaction. The same is true for conference calls. The table column *nInternToInternCalls* shows up on some of the reports as *NonAcdInbound* with the addition of *nExternToInternCalls*. Because of this, IC calculates the statistic for *NonAcdInboundCalls* from two values in the SQL tables, *nExternToInternCalls* and *nInternToInternCalls* on some of the Distribution

Queue Performance reports, The table column *NonAcdInboundCalls* include internal interactions, not just external interactions. Note that internal inbound interactions include all internal to internal calls, whether the agent placed or received the interaction. IC currently has no way of determining the direction of internal interactions, so we include outbound internal to internal interactions.

Service Levels

The Service Level statistics in the Interval Queue Data tables are based on the number of interactions answered within the specified number of seconds by the queue. You can set service levels in Interaction Administrator under the *QueuePeriodStatisticsServiceThresholds* server parameter.

The setting for *QueuePeriodStatisticsServiceThresholds* is a list of queue service level thresholds consisting of an increasing sequence of seconds separated by commas. For example, "10,20,30,40,50,60" indicates service level thresholds set at 10, 20, 30, 40, 50, and 60 seconds. This is the default setting for IC. Several of the standard reports also depend on this default setting. If you need to change the settings for your service levels, you might also need to change any reports that use the service levels.

Also, you should keep in mind that changing the service level setting is not dynamic and does not alter data already stored in the tables. For example, if you change the service levels today, you will not be able to run reports for last week, or anytime in the past, and get consistent information. This is a result of the way IC stores service level counts in the database. IC stores them according to the sequence and not according to the sequence value. If the sequence changes, then the values the system has already stored no longer correspond to the correct setting.

Agent Queue Period Statistics Interval Table

In IC, all Agent/Users have their own User Queue that is responsible for presenting interactions directed to the agent/user. Interactions on the user queue appear in the **My Interactions** box on the **Interactions** page of the Interaction Client. The Agent Queue Statistics Interval table contains the interval interaction statistics associated with the Agent/Users User Queue. For more information on agent state and status reporting unrelated to the QPS interval, see section on "Agent Activity Table" earlier in this document.

StatServer's automated logging inserts data into this table. The system can generate multiple records for each agent during the same interval.

IC adds new records as:

- One for each distribution queue the agent is a member of.
- One record for the special "*" or summary report group that represents the sum of all interaction activity for the Agent during the interval.
- One for each custom report group, if any, assigned to interactions that are handled by the agent.

You might notice that some of the columns present in the Agent Queue Statistics table are not used. This is due to the initial design for IC reporting tables in earlier product versions. The columns remain to minimize the changes and lessen the impact on custom reports. For example, there are service level columns for agents, but we do not use them often since service level is usually not meaningful relative to a user queue.

This table is structurally similar to IWrkgrpQueueStats and contains *tAgent* values. For more information on comparing interval queue data, see the section on "Interval Queue Data" earlier in the document.

cType Hierarchy

The Agent Queue Statistics Interval log normally only contains data defined for the *cType = 'A'* hierarchy. This hierarchy is defined by the following attributes associated with an interaction—in order— *'Agent'*|*'Distribution Queue'* | *'Report Group'* | *'Interaction Type'.*

This would mean for *cType = 'A'*, that *cName* is the user or agent taking the interaction. *cReportGroup* is the system-assigned distribution queue name. *cHKey3* is a report group that IC assigns to an interaction and *cHKey4* is the Interaction Type. The single asterisk, "*", in a column represents that this record applies to the ALL or the summary group of the previous column. The single dash, "-", in a column represent a null or blank value for the column. For more information, see the Record Type Hierarchy section in the Interval Queue Data summary.

Examples of values:

cType	cName	cReportGroup	cHKey3	cHKey4	Explanation
A	Agent1	*	*	*	Agent1 summary
A	Agent 1	Queue 1	*	*	User1 by Queue1 summary
A	Agent 1	-	-	Call	User1 without report group, but subdivided by interaction media type.
A	Agent 1	Queue 1	-	Email	User1 by Queue1 without report group, but sub-divided by email interaction type.
A	Agent 1	Queue 2	-	Chat	User1 by Queue2 without report group, but subdivided by chat interaction type.
A	Agent 2	Queue 2	*	*	User2 by Queue2 summary
A	Agent 2	Queue 2	DNIS-4444	*	User2 by Queue2 and report group, DNIS-4444, summary
A	Agent 2	Queue 2	DNIS-4444	Call	User2 by Queue2 and report group, DNIS-4444, and call interaction type

Physical Attributes

Log Identifier	Log Name	Table Name
90	Agent Queue Statistics Interval Log	IAgentQueueStats

Column Definitions

Column Name (Bold=Indexed)	IC Data Type	SQL Data Type (Size)	Description
cName	String	Varchar (50)	Name of the Agent/User queue that originated these statistics.
cReportGroup	String	Varchar (50)	Subgroup for cName. Agent/User queues automatically have a report group for each distribution queue that they are a member of. A special '*' report group represents the sum of all activity for the cName in this interval. Any custom report groups that may be assigned to interactions that the agent handles are also automatically added to the records associated with the cName.
cType	String	Char (1)	Type of Queue statistics data. In the case of the Agent Queue Statistics Interval table, an **A** will be entered to signify the type is Agent/User.
dIntervalStart	DateTime	DateTime (8)	Starting date/time of the interval. The interval is a server parameter value setup in IA. The default interval is 1800 seconds (30 minutes) and is always relative to the hour. In some cases, the interval start times might not be exactly on the hour. This is usually due to stopping and starting IC. These odd interval start times

Column Name (Bold=Indexed)	IC Data Type	SQL Data Type (Size)	Description
			may be visible in the reports. See Appendix B for more information.
nDuration	Integer	Int (4)	The duration in seconds of the interval. If dIntervalStart is an odd start time, the duration of the interval will have a non standard length.
nEnteredAcd	Integer	Int (4)	The number of ACD related queue interactions that entered this queue during the interval. An interaction is considered ACD related if it has been operated upon by any of the ACD interaction handling tools. nEnteredAcd includes interactions whose end condition is not yet determined. Any interactions that are not answered or abandoned in the current interval might be answered or might abandon in the next interval.
nAbandonedAcd	Integer	Int (4)	The number of ACD related queue interactions that abandoned this queue during the interval. See the section above for more information on abandons. Interactions that transfer from one distribution queue to another distribution queue, where no agent ever answers the interaction in the

Column Name (Bold=Indexed)	IC Data Type	SQL Data Type (Size)	Description
			first distribution queue, will not show up in nAbandonedACD or nTransferACD. They will only show in nFlowOutAcd.
nGrabbedAcd	Integer	Int (4)	This value is not currently supported.
nLocalDisconnectAcd	Integer	Int (4)	Number of ACD queue interactions that were disconnected locally. A local disconnect is defined as a disconnect by an IC agent/user or the IC system. A remote disconnect comes from the external party or phone company.
nAlertedAcd	Integer	Int (4)	Number of ACD related queue interactions that were in an *Alerting* state while in this queue Note that interactions which transition to Alerting more than once are only counted once.
nAnsweredAcd	Integer	Int (4)	Number of ACD related queue interactions that were answered by the agent. Answered interactions are interactions that reached a Client_Connected state with an agent. This number can exceed nEnteredACD for the agent's queue because interactions can enter the queue in the previous interval and then be answered in the current interval.

Column Name (Bold=Indexed)	IC Data Type	SQL Data Type (Size)	Description
			Interactions are only answered once relative to the agent queue assignment.
nAnswered	Integer	Int (4)	Number of all interactions—ACD and non-ACD—that were answered by the agent. Answered interactions are interactions that reached a Client_Connected state with an agent. See nAnsweredAcd for more information.
nAcdSvcLvl	Integer	Int (4)	Number of seconds in the first service level. This column does not usually apply for agents. See nAnsweredAcdSvcLvl1 for more info.
nAnsweredAcdSvcLvl1	Integer	Int (4)	The number of ACD interactions answered in the first service level. This column does not usually apply for agents. The value can be applicable, but is important to remember that the value is relative to the agents queue and not the distribution queue. It can be used to track the time to answer an interaction during the ring time. In this case, the service level settings will need to be very short to capture data. For more information, see the description of service levels in the Interval Queue Data summary section above.

Column Name (Bold=Indexed)	**IC Data Type**	**SQL Data Type (Size)**	**Description**
nAnsweredAcdSvcLvl2	Integer	Int (4)	The number of ACD interactions answered in the second service level. This column does not usually apply for agents. See nAnsweredAcdSvcLvl1 for more information.
nAnsweredAcdSvcLvl3	Integer	Int (4)	The number of ACD interactions answered in the third service level. This column does not usually apply for agents. See nAnsweredAcdSvcLvl1 for more information.
nAnsweredAcdSvcLvl4	Integer	Int (4)	The number of ACD interactions answered in the fourth service level. This column does not usually apply for agents. See nAnsweredAcdSvcLvl1 for more information.
nAnsweredAcdSvcLvl5	Integer	Int (4)	The number of ACD interactions answered in the fifth service level. This column does not usually apply for agents. See nAnsweredAcdSvcLvl1 for more information.
nAnsweredAcdSvcLvl6	Integer	Int (4)	The number of ACD interactions answered in the sixth service level. This column does not usually apply for agents. See nAnsweredAcdSvcLvl1 for more information.
nAbandonAcdSvcLvl1	Integer	Int (4)	The number of ACD interactions abandoned in the first service level. This column does not usually apply for agents. See

Column Name (Bold=Indexed)	IC Data Type	SQL Data Type (Size)	Description
			nAnsweredAcdSvcLvl1 for more information.
nAbandonAcdSvcLvl2	Integer	Int (4)	The number of ACD interactions abandoned in the second service level. This column does not usually apply for agents. See nAnsweredAcdSvcLvl1 for more information.
nAbandonAcdSvcLvl3	Integer	Int (4)	The number of ACD interactions abandoned in the third service level. This column does not usually apply for agents. See nAnsweredAcdSvcLvl1 for more information.
nAbandonAcdSvcLvl4	Integer	Int (4)	The number of ACD interactions abandoned in the fourth service level. This column does not usually apply for agents. See nAnsweredAcdSvcLvl1 for more information.
nAbandonAcdSvcLvl5	Integer	Int (4)	The number of ACD interactions abandoned in the fifth service level. This column does not usually apply for agents. See nAnsweredAcdSvcLvl1 for more information.
nAbandonAcdSvcLvl6	Integer	Int (4)	The number of ACD interactions abandoned in the sixth service level. This column does not usually apply for agents. See nAnsweredAcdSvcLvl1 for more information.

Column Name (Bold=Indexed)	IC Data Type	SQL Data Type (Size)	Description
tGrabbedAcd	Integer	Int (4)	This value is not currently supported.
tAnsweredAcd	Integer	Int (4)	The sum of the time in seconds of all ACD interactions that were in queue before entering a Client_Connected state.
mtAnsweredAcd	Integer	Int (4)	The maximum time in seconds that an interaction was in the queue before entering a Client_Connected state.
tAbandonedAcd	Integer	Int (4)	The sum of the time in seconds all abandoned ACD interactions were in queue before they abandoned.
tTalkAcd	Integer	Int (4)	The sum of the time, in seconds, all ACD interactions spent from when they first entered a Client_Connected state until the time the ACD interactions went inactive or flowed out of the queue. It is possible to have talk time appear in an interval without having any new ACD interactions for that interval. The time comes from ACD interactions that connected in that interval. Interactions might have entered the queue in the previous interval. It is possible to have talk time on a chat. Think of tTalkAcd as the time an interaction

Column Name (Bold=Indexed)	IC Data Type	SQL Data Type (Size)	Description
			is active with an agent
nHoldAcd	Integer	Int (4)	Number of ACD interactions that were placed on hold while in this queue.
tHoldAcd	Integer	Int (4)	The sum of the time,in seconds, all ACD interactions spent on hold while in this queue.
tAcw	Integer	Int (4)	The sum of the time,in seconds, the agent spent in an After Contact Work status, also known as wrap up time. The count starts when an interaction goes inactive, usually due to a local or remote disconnect, and ends when the agent leaves the After Contact Work status. This value might—or might not—include non-ACD interactions depending on how the statuses are setup.
nExternToInternCalls	Integer	Int (4)	Number of interactions, ACD and non-ACD, originating from external locations and connecting to internal extensions.
nExternToInternAcdCalls	Integer	Int (4)	Number of ACD interactions from external locations to internal extensions.
nInternToExternCalls	Integer	Int (4)	Number of all interactions from internal extension to external locations.

Column Name (Bold=Indexed)	IC Data Type	SQL Data Type (Size)	Description
nInternToExternAcdCalls	Integer	Int (4)	Number of ACD interactions from internal extension to external locations.
nInternToInternCalls	Integer	Int (4)	Number of all interactions from internal extensions to internal extensions. Internal to internal calls are counted as inbound calls whether the agent placed or received the call because IC has no way of determining the direction of internal calls.
nInternToInternAcdCalls	Integer	Int (4)	Number of ACD interactions from internal extensions to internal extensions. This includes ACD Calls transferred between agents in the same distribution queue. Both agents are credited with the ACD call.
tExternToInternCalls	Integer	Int (4)	Sum of seconds for all interactions from external locations to internal extensions.
tExternToInternAcdCalls	Integer	Int (4)	Sum of seconds for ACD interactions from external locations to internal extensions.
tInternToExternCalls	Integer	Int (4)	Sum of seconds for all interactions from internal extensions to external locations.
tInternToExternAcdCalls	Integer	Int (4)	Sum of seconds for ACD interactions from internal extensions to external locations.

Column Name (Bold=Indexed)	**IC Data Type**	**SQL Data Type (Size)**	**Description**
tInternToInternCalls	Integer	Int (4)	Sum of seconds for all interactions from internal extensions to internal extensions.
tInternToInternAcdCalls	Integer	Int (4)	Sum of seconds for ACD interactions from internal extensions to internal extensions. This includes ACD Interactions transferred between agents in the same distribution queue. Both agents are credited with the ACD interaction.
nAcwCalls	Integer	Int (4)	Number of outbound interactions made by the agent during After Contact Work time. If the agent places an interaction after handling an ACD interaction, that interaction, or any other interaction placed during the ACW time, is considered a nACWcall. nACWCall is also accumulated when an agent puts an ACD interaction on hold and places a consult call with another party. IC assumes that any call placed with an ACD interaction on hold is nACWCall activity. IC actually caches the last set of containers associated with the agent's last ACD interaction. This cache is reset any time the agent answers another ACD interaction. When IC

Column Name (Bold=Indexed)	IC Data Type	SQL Data Type (Size)	Description
			sees an ACD interaction or conditions on the agent queue that would count as an AcwCall, it processes the time and the count against the cached container list. It is done this way because the ACD interaction that made this association might actually be dead and gone from the system by the time the AcwCall event occurs.
tAcwCalls	Integer	Int (4)	Sum of time, in seconds, the agent spent on outbound interactions during After Call Work time. Also see nAcwCalls.
nTransferedAcd	Integer	Int (4)	Number of ACD interactions transferred while on this queue. The destination could have been another agent in the same distribution queue, or an agent outside of the distribution queue. Calls that transfer from one distribution queue to another distribution queue will not show up in nTransferACD. They will only show in nFlowOutAcd. Only set for Call interaction types.
nNotAnsweredAcd	Integer	Int (4)	Number of ACD interactions that were not answered when presented to the agent as an Alerting

Column Name (Bold=Indexed)	IC Data Type	SQL Data Type (Size)	Description
			interaction. nNotAnsweredAcd interactions will also count towards nFlowOutAcd
tAlertedAcd	Integer	Int (4)	Sum of the time, in seconds, ACD interactions spent in an Alerting state on this user queue. Also referred to as Ring time.
nFlowOutAcd	Integer	Int (4)	The number of ACD interactions that flowed out during this interval. Flow outs are defined as queue interactions that were removed from a queue without reaching an inactive state. Queue interactions have an inactive state that is used to mark the interaction for deletion by the ACD subsystem after the interaction has either been answered or abandoned. The most common reason for an interaction to not reach an inactive state, and thus flow out, would be a queue interaction that is transferred. Also, nNotAnsweredAcd interactions will count towards nFlowOutAcd. Another possible reason for a flow out to occur would be from hunt group alerts. A multiple alert for hunt group places the same interaction

Column Name (Bold=Indexed)	**IC Data Type**	**SQL Data Type (Size)**	**Description**
			on multiple user/agent queues. The interaction is considered a flow out in each of the user queues were the users do not answer. This would mean a large number of flow outs could be generated for each member of the hunt group when using this type of interaction routing.
tFlowOutAcd	Integer	Int (4)	Sum of seconds ACD interactions were in queue before being counted in nFlowOutAcd. See nFlowOutAcd for more information.
NStartWaitAlertAcdCalls	Integer	Int (4)	Number of ACD interactions that were waiting to be answered or were alerting to be answered at the start of the interval.
nStartActiveAcdCalls	Integer	Int (4)	Number of ACD interactions that were active with the agent at start of the interval.
nStartHeldAcdCalls	Integer	Int (4)	Number of ACD interactions that were held at start of the interval.
nEndWaitAlertAcdCalls	Integer	Int (4)	Number of ACD interactions that are waiting to be answered or were alerting to be answered at the end of the interval.

Column Name (Bold=Indexed)	IC Data Type	SQL Data Type (Size)	Description
nEndActiveAcdCalls	Integer	Int (4)	Number of ACD interactions that are active with the agent at the end of the interval.
nEndHeldAcdCalls	Integer	Int (4)	Number of ACD interactions that are held at the end of the interval.
nTransferWithinAcdCalls	Integer	Int (4)	Currently not implemented in this release.
nTransferOutAcdCalls	Integer	Int (4)	Currently not implemented in this release.
nDisconnectAcd	Integer	Int (4)	Number of ACD interactions that disconnected while still on the queue. A flow out would not be counted. The main utility of this value is in validating that all interactions that entered the queue were counted correctly.
tAgentLoggedIn	Integer	Int (4)	Sum of the time, in seconds, the agent was logged in to the client. Not the same as being in an ACD logged in state.

Column Name (Bold=Indexed)	**IC Data Type**	**SQL Data Type (Size)**	**Description**
tAgentAvailable	Integer	Int (4)	Sum of time, in seconds, agents were in an available status, whether or not the agent is activated. This column relates specifically to the agent and not the interaction type, meaning it is not limited to ACD-available statuses or ACD workgroups. Ring time is currently included in the *tAgentAvailable* time.
tAgentTalk	Integer	Int (4)	Sum of time, in seconds, the agent was on interactions (ACD and non-ACD) from first Client_Connected until end of ACW for this queue.
tAgentOtherBusy	Integer	Int (4)	Sum of the time, in seconds, the agent was working on interactions (ACD and non-ACD) for queues other this one.
tAgentOnAcdCall	Integer	Int (4)	Sum of the time, in seconds the agent was working on ACD interactions for this queue only.
tAgentOnOtherAcdCall	Integer	Int (4)	Sum of the time, in seconds, the agent was working on ACD interactions for other queues.
tAgentInAcw	Integer	Int (4)	Sum of the time, in seconds the agent was in an After Contact Work state.

Column Name (Bold=Indexed)	IC Data Type	SQL Data Type (Size)	Description
tAgentOnNonAcdCall	Integer	Int (4)	Sum of the time, in seconds the agent was working on non-ACD interactions.
tAgentDnd	Integer	Int (4)	Sum of the time, in seconds, the agent was in a Do Not Disturb state. Also, tAgentDND is part of a set of values that must always sum up to tAgentLoggedIn. It is driven by the combination of ACD availability and client status.
tAgentNotAvailable	Integer	Int (4)	Sum of the time, in seconds, the agent was not available to take ACD interactions, but was logged in to the system.
tAgentAcdLoggedIn	Integer	Int (4)	Sum of the time, in seconds, the agent was logged in, activated in the queue, and available to take interactions in a status that has the **Status allows ACD calls** box checked. By default, only the Available status has this box checked. All three conditions must be met to count toward tAgentAcdLoggedIn time. This column does not count the time an agent is actively engaged in ACD interactions, for example, talking on ACD calls.

Column Name (Bold=Indexed)	**IC Data Type**	**SQL Data Type (Size)**	**Description**
tAgentStatusDnd	Integer	Int (4)	Sum of the time, in seconds, the agent was in a Do Not Disturb status as determined only by the current client status. If the current status is marked as DND, then DND time is accumulated. The value can differ from tAgentDND. If agents are allowed to set their status to a DND status while on an ACD interaction, then they still accumulate time in tAgentOnAcdCall, but their status is DND, so they will also accumulate tAgentStatusDND.
tAgentStatusAcw	Integer	Int (4)	Sum of the time, in seconds, the agent was in an After call Work status as determined only by the current client status. If the current status is marked as ACW, then ACW time is accumulated. The value can differ from tAgentInAcw. If agents are allowed to set their status to an ACW status while on an ACD interaction, then they still accumulate time in tAgentOnAcdCall, but their status is ACW, so they will also accumulate tAgentStatusAcw.

Column Name (Bold=Indexed)	IC Data Type	SQL Data Type (Size)	Description
tAgentLoggedInDiluted	Integer	Int (4)	Not meaningful relative to an agent queue because an agent cannot be a member of more than one agent queue. See IWrkgrpQueueStats.
tStatusGroupFollowup	Integer	Int (4)	Sum of the time, in seconds, the agent was in any status that belongs to the status group Followup. A status group is any grouping of agent status messages. There are five predefined groups in IC: Available, Break, Followup, Training, and Unavailable. Beyond just grouping statuses, status groups provide a way to track specific time in a status as part of the interval information of an agent or distribution queue. See the StatusGroup column in the AgentActivityLog table for more information. It is possible to define your own custom groups. See the online help in Interaction Administrator for more information. It is also possible to make the group to status mapping a one to one mapping. No grouping is actually forced, just encouraged.

Column Name (Bold=Indexed)	**IC Data Type**	**SQL Data Type (Size)**	**Description**
tStatusGroupBreak	Integer	Int (4)	Sum of time, in seconds, the agent was in any status that belongs to the status group Break. See tStatusGroupFollowup for more information.
tStatusGroupTraining	Integer	Int (4)	Sum of time, in seconds, the agent was in any status that belongs to the status group Training. See tStatusGroupFollowup for more information.
CustomValue1	Integer	Int (4)	Custom value for customer use.
CustomValue2	Integer	Int (4)	Custom value for customer use.
CustomValue3	Integer	Int (4)	Custom value for customer use.
CustomValue4	Integer	Int (4)	The default value is set to mtAbandonedACD. The mtAbandonedACD is the maximum amount of time a customer waited before they abandoned in an interval. If a customer is using this custom value for their own purpose their customizations will override this default setting and the customizations will continue to work as configured.
CustomValue5	Integer	Int (4)	The default value is set to nMessageACD. The NMessageACD is the number of ACD interactions that went to a message state, voice mail for

Column Name (Bold=Indexed)	IC Data Type	SQL Data Type (Size)	Description
			interactions, during the interval. Please note that subtracting this value from nAbandonedACD will not give you a true voice mail count, since it is possible for interactions to go in and out of voice mail several times, or for an interaction to be answered and then sent to voice mail. If a customer is using this custom value for their own purpose their customizations will override this default setting and the customizations will continue to work as configured.
CustomValue6	Integer	Int (4)	The default value is set to nRequestedSuperAssistACD. NRequestSuperAssistACD is the number of supervisor-assist requests that were placed from the clients for ACD interaction during the interval. If a customer is using this custom value for their own purpose their customizations will override this default setting and the customizations will continue to work as configured.

Column Name (Bold=Indexed)	IC Data Type	SQL Data Type (Size)	Description
I3TimeStampGMT	(System Supplied)	DateTime (8)	System supplied date time when the row was sent to the IC Logging server for insertion into the table.
SiteId	Integer	SmallInt (2)	Site identifier of the source for this record. Used in multi-site installations to distinguish the origin of the record. Configured when setting up IC.
SubSiteId	Integer	SmallInt (2)	Currently not used. The purpose of this field is to divide use within a site, such as for multiple independent companies on one IC system. Until IC has such functionality, this field is zero and is reserved for future use.
cHKey3	String	Varchar (50)	This represents the third level of hierarchy grouping in terms of reporting. The most common value for this field will be the interaction type. This field will have a default value of "*". For more information, see the record type hierarchy section in the Interval Queue Data summary.
cHKey4	String	Varchar (50)	This represents the fourth level of hierarchy grouping in terms of reporting. This field will not be used in a typical implementation.

Column Name (Bold=Indexed)	IC Data Type	SQL Data Type (Size)	Description
			This field will have a default value of "*". For more information, see the record type hierarchy section in the Interval Queue Data summary.

Index Definitions

Index Name	Index Type	Column Name	Order
I3TimeStampGMT	Clustered non-unique	I3TimeStampGMT	Ascending
		SiteId	Ascending
		SubSiteId	Ascending
PKQueueName	Primary Key	cName	Ascending
		cReportGroup	Ascending
		cHKey3	Ascending
		cHKey4	Ascending
		cType	Ascending
		I3TimeStampGMT	Ascending
		dIntervalStart	Ascending
		SiteId	Ascending
		SubSiteId	Ascending
IEIntervalStart	Non Unique	dIntervalStart	Ascending
		cName	Ascending
		SiteId	Ascending
		SubSiteId	Ascending

Workgroup Queue Statistics Interval Table

The Workgroup Queue Statistics Interval table contains interval-based interaction statistics associated with distribution queues. The automated StatServer's logging inserts data into this table for each workgroup/queue for which there was activity or agents logged in. IC might generate multiple records for each distribution queue during the same interval. IC adds new records to this table in this way:

- One for each custom report group, if any, IC assigns to interactions that are handled by the distribution queue.
- One record for the special "*" or summary report group that represents the sum of all interaction activity for the distribution queue during the interval.

You might notice that some of the columns present in the Workgroup Queue Statistics table are not used. This is due to the initial design for IC reporting tables in earlier product versions. The columns remain to minimize the changes and lessen the impact on custom reports.

This table is structurally similar to IAgentQueueStats and contains *tAgent* values. See the Interval Queue Data summary, earlier in this document, for more information on comparing interval queue data.

cType Hierarchy

The Workgroup Queue Statistics Interval table normally only contains data defined for the *cType = 'W'* hierarchy. This hierarchy is defined by the following attributes associated with an interaction (in order), *'Distribution Queue'* | *'Report Group'* | *'Interaction Type'.*

This means for *cTyp*e = 'S', that *cName* is the statistics group name. *cReportGroup* is any report group IC assigns to an interaction. IC uses *cHKey*3 for the interaction type. IC does not use *cHKey4*, so it will always be "*".The single asterisk, "*", in a column represents that this record applies to the ALL or the summary group of the previous column. The single dash, "-", in a column represent a null or blank value for the column. For more information, see "Record Type Hierarchy" earlier in this document.

Examples of values:

cType	cName	cReportGroup	cHKey3	cHKey4	Explanation
W	Queue1	*	*	*	Queue1 summary
W	Queue1	-	Call	*	Queue1 without a report group, but subdivided by call interaction type.
W	Queue1	-	Chat	*	Queue1 without a report group but subdivided by chat interaction type.
W	Queue2	*	*	*	Queue2 summary
W	Queue2	DNIS-4444	*	*	Queue2 by report group, DNIS-4444, summary

cType	cName	cReportGroup	cHKey3	cHKey4	Explanation
W	Queue2	DNIS-4444	Call	*	Queue2 by report group, DNIS-4444, and call interaction type.

Physical Attributes

Log Identifier	Log Name	Table Name
92	Workgroup Queue Statistics Interval	IwrkgrpQueueStats

Column Definitions

Column Name (Bold=Indexed)	IC Data Type	SQL Data Type (Size)	Description
CName	String	Varchar (50)	Name of the distribution queue that originated these statistics.
CReportGroup	String	Varchar (50)	Subgroup for *cName*. A special '*' report group represents the sum of all activity for the *cName* in this interval. Any custom report groups that might be assigned to interactions that the distribution queue handles are also automatically added to the records associated with the *cName*.
CType	String	Char (1)	Type of queue statistics data. In the case of the Workgroup Queue Statistics Interval table, a "W" will be entered to signify the type is a distribution queue.

Column Name (Bold=Indexed)	IC Data Type	SQL Data Type (Size)	Description
DIntervalStart	DateTime	DateTime (8)	Starting date/time of the interval. The interval is a server parameter value setup in IA. The default interval is 1800 seconds (30 minutes) and is always relative to the hour. In some cases, the interval start times might not be exactly on the hour. This is usually due to stopping and starting IC. These odd interval start times might be visible in the reports. See Appendix B for more information.
nDuration	Integer	Int (4)	Duration in seconds of the interval. If *dIntervalStart* is an odd start time, the duration of the interval will have a non standard length.
nEnteredAcd	Integer	Int (4)	The number of ACD related queue interactions that entered this queue during the interval. An interaction is considered ACD related if it has been operated upon by any of the ACD interaction handling tools. *nEnteredAcd* includes interactions whose end condition is not yet determined. Any interactions that are not answered or abandoned in the current interval might be answered or might abandon in the next interval. An interaction is only counted as having entered once relative to this queue assignment.
nAbandonedAcd	Integer	Int (4)	The number of ACD related queue interactions that abandoned this queue during the interval. Interactions that transfer from one Type of queue statistics data. In the case of the Workgroup Queue Statistics Interval table, a "W" will be entered to signify the type is a distribution queue. To

Column Name (Bold=Indexed)	IC Data Type	SQL Data Type (Size)	Description
			another distribution queue (where no agent ever answers the interaction in the first distribution queue) will not show up in *nAbandonedACD* or *nTransferACD*. They will only show in *nFlowOutAcd*. See the section in the Interval Queue Data summary section earlier in this document for more information on abandons.
nGrabbedAcd	Integer	Int (4)	This value is not currently supported.
nLocalDisconnectAcd	Integer	Int (4)	Number of ACD queue interactions that were disconnected locally. A local disconnect is defined as a disconnect by an IC agent/user or the IC system. A remote disconnect comes from the external party or phone company.
nAlertedAcd	Integer	Int (4)	Number of ACD related queue interactions that were in an "Alerting" state while in this queue (interactions which transition to "Alerting" more than once are only counted once).
nAnsweredAcd	Integer	Int (4)	Number of ACD related queue interactions that were answered by agents that were members of this queue. Answered interactions are interactions that reached a "Client_Connected" state with an agent. An interaction is only counted as answered once relative to this queue assignment. If the interaction is transferred to another agent within the same distribution queue and is answered by another agent, it is still only counted once. Voice mail interactions are considered abandons and do not count towards *nAnsweredAcd*. This

Column Name (Bold=Indexed)	IC Data Type	SQL Data Type (Size)	Description
			value can exceed *nEnteredACD* for a distribution queue because interactions can enter the queue in the previous interval and be answered in the current interval.
nAnswered	Integer	Int (4)	Number of all interactions (ACD and non-ACD) that were answered by the agents that were members of this queue. Answered interactions are interactions that reached a “Client_Connected” state with an agent. See *nAnsweredAcd* for more information.
nAcdSvcLvl	Integer	Int (4)	Number of seconds in the first service level. For more information, see the description of service levels in the Interval Queue Data summary section above.
nAnsweredAcdSvcLvl1	Integer	Int (4)	Number of ACD interactions answered in first service level. The time tracked begins from the point the interaction becomes an ACD interaction, which is not necessarily the same as when it enters the queue. For more information, see the description of service levels in the Interval Queue Data summary section earlier in this document.
nAnsweredAcdSvcLvl2	Integer	Int (4)	Number of ACD interactions answered in second service level. See *nAnsweredAcdSvcLvl1*.
nAnsweredAcdSvcLvl3	Integer	Int (4)	Number of ACD interactions answered in third service level. See *nAnsweredAcdSvcLvl1*.
nAnsweredAcdSvcLvl4	Integer	Int (4)	Number of ACD interactions answered in fourth service level. See *nAnsweredAcdSvcLvl1*.
nAnsweredAcdSvcLvl5	Integer	Int (4)	Number of ACD interactions answered in fifth service level.

Column Name (Bold=Indexed)	IC Data Type	SQL Data Type (Size)	Description
			See *nAnsweredAcdSvcLvl1*.
nAnsweredAcdSvcLvl6	Long	Int (4)	Number of ACD interactions answered in sixth service level. See *nAnsweredAcdSvcLvl1*.
nAbandonAcdSvcLvl1	Integer	Int (4)	Number of ACD interactions abandoned in first service level. The time tracked begins from the point the interaction becomes an ACD interaction; which is not necessarily the same as when it enters the queue. For more information, see the description of service levels and abandoned interactions in the Interval Queue Data summary section above.
nAbandonAcdSvcLvl2	Integer	Int (4)	Number of ACD interactions abandoned in second service level. See *nAbandonAcdSvcLvl1*.
nAbandonAcdSvcLvl3	Integer	Int (4)	Number of ACD interactions abandoned in third service level. See *nAbandonAcdSvcLvl1*.
nAbandonAcdSvcLvl4	Integer	Int (4)	Number of ACD interactions abandoned in fourth service level. See *nAbandonAcdSvcLvl1*.
nAbandonAcdSvcLvl5	Integer	Int (4)	Number of ACD interactions abandoned in fifth service level. See *nAbandonAcdSvcLvl1*.
nAbandonAcdSvcLvl6	Integer	Int (4)	Number of ACD interactions abandoned in Sixth service level. See *nAbandonAcdSvcLvl1*.
tGrabbedAcd	Integer	Int (4)	This value is not currently supported.

Column Name (Bold=Indexed)	**IC Data Type**	**SQL Data Type (Size)**	**Description**
tAnsweredAcd	Integer	Int (4)	The sum of the time, in seconds, of all ACD interactions that were in queue before entering “Client_Connected” state. The time tracked begins from the point the interaction becomes an ACD interaction, which is not necessarily the same as when it enters the queue.
mtAnsweredAcd	Integer	Int (4)	The maximum time, in seconds, an ACD interaction was in the queue before entering a “Client_Connected” state. The time tracked begins from the point the interaction becomes an ACD interaction, which is not necessarily the same as when it enters the queue.
tAbandonedAcd	Integer	Int (4)	The sum of the time, in seconds, all abandoned ACD interactions were in queue before they abandoned during this interval. The time tracked begins from the point the interaction becomes an ACD interaction, which is not necessarily the same as when it enters the queue.

Column Name (Bold=Indexed)	**IC Data Type**	**SQL Data Type (Size)**	**Description**
tTalkAcd	Integer	Int (4)	The sum of the time, in seconds, all ACD interactions spent from when they first entered a "Client_Connected" state until the time the ACD interactions went inactive or flowed out of the queue. It is possible to have talk time appear in an interval without having any new ACD interactions for that interval. The time comes from ACD interactions that connected in that interval. Interactions might have entered the queue in the previous interval. It is possible to have talk time on a chat. Think of *tTalkAcd* as the time an interaction is active with an agent
nHoldAcd	Integer	Int (4)	Number of ACD related interactions that were placed on hold while in this queue.
tHoldAcd	Integer	Int (4)	The sum of the time, in seconds, all ACD interactions spent on hold while in this queue.
tAcw	Integer	Int (4)	The sum of the time, in seconds, agents spent in an "After Contact Work" status (also known as wrap up time). The count starts when an interaction goes inactive (usually due to a local or remote disconnect) and ends when the agent leaves the "After Call Work" status. This value may or may not include non-ACD interactions depending on how the statuses are setup.
nExternToInternCalls	Integer	Int (4)	Number of interactions (ACD and non-ACD) originating from external locations and connecting to internal extensions.

Column Name (Bold=Indexed)	IC Data Type	SQL Data Type (Size)	Description
nExternToInternAcdCalls	Integer	Int (4)	Number of ACD interactions from external locations to internal extensions.
nInternToExternCalls	Integer	Int (4)	Number of all interactions from internal extension to external locations.
nInternToExternAcdCalls	Integer	Int (4)	Number of ACD interactions from internal extension to external locations.
nInternToInternCalls	Integer	Int (4)	Number of all interactions from internal extensions to internal extensions. Internal to internal calls are counted as inbound calls whether an agent placed or received the call because IC has no way of determining the direction of internal calls.
nInternToInternAcdCalls	Integer	Int (4)	Number of ACD interactions from internal extensions to internal extensions. This includes ACD Interactions transferred between agents in the same distribution queue. Both agents are credited with an ACD interaction.
tExternToInternCalls	Integer	Int (4)	Sum of seconds for all interactions from external locations to internal extensions.
tExternToInternAcdCalls	Integer	Int (4)	Sum of seconds for ACD interactions from external locations to internal extensions.
tInternToExternCalls	Integer	Int (4)	Sum of seconds for all interactions from internal extensions to external locations.
tInternToExternAcdCalls	Integer	Int (4)	Sum of seconds for ACD interactions from internal extensions to external locations. (Not Implemented or used in current version, but it is set when using Interaction Dialer.)

Column Name (Bold=Indexed)	**IC Data Type**	**SQL Data Type (Size)**	**Description**
tInternToInternCalls	Integer	Int (4)	Sum of seconds for all interactions from internal extensions to internal extension.
tInternToInternAcdCalls	Integer	Int (4)	Sum of seconds for ACD interactions from internal extensions to internal extensions. This includes ACD Interactions transferred between agents in the same distribution queue. Both agents are credited with an ACD interaction.
nAcwCalls	Integer	Int (4)	Number of outbound interactions made by agents during “After Contact Work” time. If an agent places an interaction after handling an ACD interaction, that interaction, or any other interaction placed during the ACW time, is considered a *nACWcall*. nACWCall is also accumulated when an agent puts an ACD interaction on hold and places a consult call with another party. It is assumed that any call placed with an ACD interaction on hold is nACWCall activity. IC actually caches the last set of containers associated with the agent's last ACD interaction. This cache is reset any time the agent answers another ACD interaction. When IC sees ACD or conditions on the agent that would count as an AcwCall, it processes the time and the count against the cached container list. It is done this way because the ACD interaction that made this association might actually be dead and gone from the system by the time the *AcwCall* event occurs.

Column Name (Bold=Indexed)	**IC Data Type**	**SQL Data Type (Size)**	**Description**
tAcwCalls	Integer	Int (4)	Sum of time, in seconds, the agent spent on outbound interactions during “After Contact Work” time. Also see *nAcwCalls*.
nTransferedAcd	Integer	Int (4)	Number of ACD interactions transferred while on this queue. The destination could have been another agent in the same distribution queue, or an agent outside of the distribution queue. Interactions that transfer from one distribution queue to another distribution queue will not show up in *nTransferACD*. They will only show in *nFlowOutAcd*. Only set for Call interaction types.
nNotAnsweredAcd	Integer	Int (4)	Number of ACD interactions that were not answered when presented to agents as an “Alerting” interaction.
tAlertedAcd	Integer	Int (4)	Sum of the time, in seconds, ACD interactions spent in an “Alerting” state on agent queues. Also referred to as Ring time.

Column Name (Bold=Indexed)	**IC Data Type**	**SQL Data Type (Size)**	**Description**
nFlowOutAcd	Integer	Int (4)	The number of ACD interactions that flowed out during this interval. Flow outs are defined as queue interactions that were removed from a queue without reaching an inactive state. Queue interactions have an inactive state that is used to mark the interaction for deletion by the ACD subsystem after the interaction has either been answered or abandoned. The most common reason for an interaction to not reach an inactive state, and thus flow out, would be a queue interaction that is transferred. *nNotAnsweredAcd* interactions will also count towards *nFlowOutAcd*
tFlowOutAcd	Integer	Int (4)	Sum of seconds ACD interactions were in queue before being counted in *nFlowOutAcd*. See *nFlowOutAcd* for more information. The time tracked begins from the point the interaction becomes an ACD interaction; which is not necessarily the same as when it enters the queue.
nStartWaitAlertAcdCalls	Integer	Int (4)	Number of ACD interactions that were waiting to be answered or were alerting to be answered at the start of the interval.
nStartActiveAcdCalls	Integer	Int (4)	Number of ACD interactions that were active with an agent at start of the interval.
nStartHeldAcdCalls	Integer	Int (4)	Number of ACD interactions that were held at start of the interval.

Column Name (Bold=Indexed)	**IC Data Type**	**SQL Data Type (Size)**	**Description**
nEndWaitAlertAcdCalls	Integer	Int (4)	Number of ACD interactions that are waiting to be answered or were alerting to be answered at the end of the interval.
nEndActiveAcdCalls	Integer	Int (4)	Number of ACD interactions that are active with an agent at the end of the interval.
nEndHeldAcdCalls	Integer	Int (4)	Number of ACD interactions that are held at the end of the interval.
nTransferWithinAcdCalls	Integer	Int (4)	Not implemented for this release.
nTransferOutAcdCalls	Integer	Int (4)	Not implemented for this release.
nDisconnectAcd	Integer	Int (4)	Number of ACD interactions that disconnected while still on the queue. A flow out would not be counted. The main utility of this value is in validating that all interactions that entered the queue were counted correctly.
tAgentLoggedIn	Integer	Int (4)	Sum of the time, in seconds, the agent was logged in to the client. Not the same as being in an ACD logged in state.
tAgentAvailable	Integer	Int (4)	Sum of time, in seconds, agents were in an available status, whether or not the agent is activated. This column relates specifically to the agent and not the interaction type, meaning it is not limited to ACD-available statuses or ACD workgroups. Ring time is currently included in the *tAgentAvailable* time.
tAgentTalk	Integer	Int (4)	Sum of time, in seconds, agents were on interactions (ACD and non-ACD) from first “Client_Connected” until end of ACW for this queue.

Column Name (Bold=Indexed)	IC Data Type	SQL Data Type (Size)	Description
tAgentOtherBusy	Integer	Int (4)	Sum of the time, in seconds, agents were working on interactions (ACD and non-ACD) for queues other this one.
tAgentOnAcdCall	Integer	Int (4)	Sum of the time, in seconds, agents were working on ACD interactions for this queue only.
tAgentOnOtherAcdCall	Integer	Int (4)	Sum of the time, in seconds, agents were working on ACD interactions for other queues.
tAgentInAcw	Integer	Int (4)	Sum of the time, in seconds, agents were in an "After call Work" state.
tAgentOnNonAcdCall	Integer	Int (4)	Sum of the time, in seconds, agents were working on non-ACD interactions.
tAgentDnd	Integer	Int (4)	Sum of the time, in seconds, agents were in a "Do Not Disturb" state. *tAgentDND* is part of a set of values that must always sum up to *tAgentLoggedIn*. It is driven by the combination of ACD availability and client status.
tAgentNotAvailable	Integer	Int (4)	Sum of the time, in seconds, agents were not available to take ACD interactions, but were logged in to the system.
tAgentAcdLoggedIn	Integer	Int (4)	Sum of the time, in seconds, the agent was logged in, activated in the queue, and available to take interactions in a status that has the "Status allows ACD calls" box checked. By default, only the Available status has this box checked. All three conditions of the agent must be met to count toward tAgentAcdLoggedIn time: 1. Agent must be logged in; 2. Agent must be in a status that has Status Allows ACD calls selected; 3. Agent must be activated in the workgroup

Column Name (Bold=Indexed)	**IC Data Type**	**SQL Data Type (Size)**	**Description**
			queue. This column does not count the time an agent is actively engaged in ACD interactions, for example, talking on ACD calls. Note: If given permission in IA, an agent can change his or her workgroup status in IC Client, affecting individual workgroup totals.
tAgentStatusDnd	Integer	Int (4)	Sum of the time, in seconds, agents were in a “Do Not Disturb” status as determined only by the current client status. If the current status is marked as DND, then DND time is accumulated. The value may differ from *tAgentDND*. If agents are allowed to set their status to a DND status while on an ACD interaction, then they still accumulate time in *tAgentOnAcdCall*, but their status is DND, so they will also accumulate *tAgentStatusDND*.
tAgentStatusAcw	Integer	Int (4)	Sum of the time, in seconds, agents were in an “After Contact Work” status as determined only by the current client status. If the current status is marked as ACW, then ACW time is accumulated. The value may differ from *tAgentInAcw*. If agents are allowed to set their status to an ACW status while on an ACD interaction, then they still accumulate time in *tAgentOnAcdCall*, but their status is ACW, so they will also accumulate *tAgentStatusAcw*.
nAgentLoggedInDiluted	Integer	Int (4)	The diluted logged in time, in seconds, for agents who are members of this queue. Calculated by dividing the time the agents were logged in by the number of queues the agents were a member of. For example, if this queue has 10

Column Name (Bold=Indexed)	**IC Data Type**	**SQL Data Type (Size)**	**Description**
			agents who were only members of this queue and were logged in for 1800 seconds, the value would be 18000. If 5 of these agents became members of a second queue and were logged in for the same amount of time to the second queue, the value would be 900 x 5 which is 4500 plus 9000 (for the 5 agents who were not a member of any other queue) for a total of 13500.
tStatusGroupFollowup	Integer	Int (4)	Sum of time, in seconds, agents were in any status that belongs to the status group "Followup". A Status Group is any grouping of agent status messages. There are five predefined groups in IC: **Available**, **Break**, **Followup**, **Training**, and **Unavailable**. Beyond just grouping statuses, Status Groups provide a way to track specific time in a status as part of the interval information of an agent or distribution queue. See the *StatusGroup* column in the AgentActivityLog table for more information. It is possible to define your own custom groups. See the online help in Interaction Administrator for more information. It is also possible to make the group to status mapping a one to one mapping. No grouping is actually forced, just encouraged.
tStatusGroupBreak	Integer	Int (4)	Sum of time, in seconds, agents were in any status that belongs to the status group **Break**. See *tStatusGroupFollowup* for more information.
tStatusGroupTraining	Integer	Int (4)	Sum of time, in seconds, agents were in any status that

Column Name (Bold=Indexed)	**IC Data Type**	**SQL Data Type (Size)**	**Description**
			belongs to the status group **Training**. See *tStatusGroupFollowup* for more information.
CustomValue1	Integer	Int (4)	Custom value for customer use.
CustomValue2	Integer	Int (4)	Custom value for customer use.
CustomValue3	Integer	Int (4)	Custom value for customer use.
CustomValue4	Integer	Int (4)	The default value is set to mtAbandonedACD. The mtAbandonedACD column is the maximum amount of time a customer waited before they abandoned in an interval. If a customer is using this custom value for their own purpose their customizations will override this default setting and the customizations will continue to work as configured.
CustomValue5	Integer	Int (4)	The default value is set to nMessageACD. The column NMessageACD is the number of ACD interactions that went to a message state (voice mail for interactions) during the interval. It should be noted that subtracting this value from nAbandonedACD will not give you a true voice mail count, since it is possible for interactions to go in and out of voice mail several times or for an interaction to be answered and then sent to voice mail. If a customer is using this custom value for their own purpose their customizations will override this default setting and the customizations will continue to work as configured.
CustomValue6	Integer	Int (4)	The default value is set to nRequestedSuperAssistACD. NRequestSuperAssistACD is the

Column Name (Bold=Indexed)	IC Data Type	SQL Data Type (Size)	Description
			number of supervisor-assist requests that were placed from the “clients” for ACD interaction during the interval. If a customer is using this custom value for their own purpose their customizations will override this default setting and the customizations will continue to work as configured.
I3TimeStampGMT	(System Supplied)	DateTime (8)	System supplied date time when the row was sent to the IC Logging server for insertion into the table.
SiteId	Integer	SmallInt (2)	Site identifier of the source of this interaction row. Used in multi-site rollup to distinguish the origin of data.
SubSiteId	Integer	SmallInt (2)	Not used in this version. The purpose of this field is to divide use within a site, such as for multiple independent companies on one IC system. Until IC has such functionality, this field is zero and is reserved for future use
cHKey3	String	Varchar (50)	This represents the third level of hierarchy grouping in terms of reporting. The most common value for this field will be the interaction type. This field will have a default value of "*". For more information, see the record type hierarchy section in the Interval Queue Data summary.
cHKey4	String	Varchar (50)	This represents the fourth level of hierarchy grouping in terms of reporting. This field will not be used in a typical implantation. This field will have a default value of "*". For more information, see the record type hierarchy section in the Interval Queue Data summary.

Index Definitions

Index Name	Index Type	Column Name	Order
I3TimeStampGMT	Clustered non-unique	I3TimeStampGMT	Ascending
		SiteId	Ascending
		SubSiteId	Ascending
PKQueueName	Primary Key	cName	Ascending
		cReportGroup	Ascending
		cHKey3	Ascending
		cHKey4	Ascending
		cType	Ascending
		I3TimeStampGMT	Ascending
		dIntervalStart	Ascending
		SiteId	Ascending
		SubSiteId	Ascending
IEIntervalStart	Non Unique	dIntervalStart	Ascending
		cName	Ascending
		SiteId	Ascending
		SubSiteId	Ascending

Statistics Group Queue Interval Table

Statistics groups, also refered to as STIDs, or *ST*atistic *ID*entifiers, are queue-like containers for interactions. IC uses these tables to categorize interactions for statistical purposes. IC typically uses them to track statistics for interactions on a level higher than real distribution queues. You can group together the information for interactions that traverse several queues, even though the interactions also reside on different queues. Because of this grouping, you can track the statistics of interactions regardless of their distribution queue assignments.

The statistics IC generates for interactions in statistical groups are limited to information about the interactions, and do not include the additional *tAgent* values IC generates for distribution queues and agent queues. See the Interval Queue Data summary section above for more information on Comparing Interval Queue Data.

You cannot associate an STID with agents. Therefore, the STID does not have any purely agent-related information available for collection. The system collects time agents spend on interactions, but any time an agent spends that is not directly associated with an interaction cannot be tracked by a statistics group.

As an example of how statistics groups work, imagine that your call center receives calls on three separate toll-free numbers. You want to determine how well your agents are processing these calls, so you need a way of organizing the statistics information generated by the IC Queue Manager. Each time a call enters the *SystemIVRWorkgroup* handler, you assign that call to one of three statistics groups depending upon which toll-free number the caller dialed. When Queue Manager generates statistics information about all of the calls in IC, it also generates records in the IStatsGroup table for the groups of calls belonging to each toll-free number statistics group.

Considerations when using STIDs

- An STID is a container for interactions, not agents.
- You create STIDs in handlers using the Assign Stats Group tool.
- You can change the STID assignment of an interaction at any time.
- You can make an interaction a member of multiple STIDs at one time.
- There are no stock reports for STIDs.
- STIDs behave just like queues when it comes to data collection. When the system assigns an interaction to a STID, it is just the same statistically as assigning the interaction to a queue.

For example, if IC assigns an interaction to a STID and then transfers the interaction to an agent directly (that is, not using any ACD tools) and the agent answers the interaction, this interaction now counts towards *nAnswered* and the appropriate interaction direction counter, such as *nExternToInternCalls*.

If IC assigns the interaction to an ACD queue right after it assigns it to the STID, instead of assigning it directly to an agent, the interaction counts towards the *nEnteredACD* statistic. The (now ACD) interaction then waits for an agent. This time spent waiting, beginning from the point of ACD assignment, counts toward the ACD service level statistics for the STID. When an agent answers the interaction, the interaction then counts towards *nAnsweredACD*.

If the system transfers the interaction to another agent or ACD queue while remaining assigned to the same STID, it is not counted again as answered or entered for this STID assignment.

An interaction can only reach the answered or entered state once, relative to the STID assignment. The service level values only apply to the time it takes for an interaction to be answered, or abandoned, and since the answered state can only happen once, interactions that transfer or are answered by a second agent or queue do not affect the service level values. This is just like on a distribution queue when the interaction is transferred between agents in the same queue. The talk time changes when multiple agents handle an interaction but the interaction counts do not.

As soon as the system removes an interaction from a STID, all statistics tracking the interaction relative to the STID stops. If the interaction is an active ACD interaction, talk time stops and *nFlowOutAcd* is incremented.

If you program a loop in a handler, you can assign and remove the same interaction to the same STID multiple times. If an interaction leaves and re-enters a STID, the second entry will count. There is nothing that would make the second entry different from any previous entry. The IC StatServer does not track that the interaction has been in the STID before, so IC generates all statistics as if the interaction were in this STID for the first time. However, an interaction can only reach the answered or entered state once, relative to a single STID assignment.

cType Hierarchy

The Statistics Group Interval table normally only contains data defined for the *cType = 'S'* hierarchy. This hierarchy is defined by the following attributes associated with an interaction (in order), *'Statistics Group'* | *'Report Group'* | *'Interaction Type'*.

This would mean for *cTyp*e = 'S', that *cName* is the statistics group name. *cReportGroup* is any report group assigned to an interaction. *cHKey*3 is used for the interaction type. *cHKey4* is not used, so it will always be "*". The single asterisk, "*", in a column represents that this record applies to the ALL or the summary group of the previous column. The single dash, "-", in a column represent a null or blank value for the column. For more information, see the record type hierarchy section in the Interval Queue Data summary.

Examples of values:

cType	cName	cReportGroup	cHKey3	cHkey4	Explanation
S	Statistics Group1	*	*	*	Statistics Group1 summary
S	Statistics Group1	-	Call	*	Statistics Group1 without a report group, but subdivided by call interaction type
S	Statistics Group1	-	Chat	*	Statistics Group1 without a report group but subdivided by chat interaction type.
S	Statistics Group2	*	*	*	Statistics Group2 summary
S	Statistics Group2	DNIS-4444	*	*	Statistics Group2 and report group, DNIS-4444, summary

cType	cName	cReportGroup	cHKey3	cHkey4	Explanation
S	Statistics Group2	DNIS-4444	Call	*	Statistics Group1 by report group, DNIS-4444, and call interaction type.

Physical Attributes

Log Identifier	Log Name	Table Name
91	Statistics Group Interval	IStatsGroup

Column Definitions

Column Name (Bold=Indexed)	IC Data Type	SQL Data Type (Size)	Description
cName	String	Varchar (50)	Name of the Statistics Group (STID) that originated these statistics.
cReportGroup	String	Varchar (50)	Subgroup for *cName*. A special '*' report group represents the sum of all activity for the *cName* in this interval. Any custom report groups that can be assigned to interactions before they are assigned to the STID are also automatically added to the records associated with the *cName*.
cType	String	Char (1)	Type of queue statistics data. In the case of the Statistics Group Interval table, an "S" is entered to signify the type is a STID.
dIntervalStart	DateTime	DateTime (8)	Starting date/time of the interval. The interval is a server parameter value setup in IA. The default interval is 1800 seconds (30 minutes) and is always relative to the hour. In some cases, the interval start times might not be exactly on the hour. This is usually due to stopping and starting IC. These odd interval start times might be visible in the reports. See Appendix B for more information.

Column Name (Bold=Indexed)	**IC Data Type**	**SQL Data Type (Size)**	**Description**
nDuration	Integer	Int (4)	Duration in seconds of the interval. If *dIntervalStart* is an odd start time, the duration of the interval will have a non standard length.
nEnteredAcd	Integer	Int (4)	The number of ACD related interactions that entered this STID during the interval. An interaction is considered ACD related if it has been operated upon by any of the ACD interaction handling tools. *nEnteredAcd* includes interactions whose end condition is not yet determined. Any interactions that are not answered or abandoned in the current interval might be answered or might abandon in the next interval. An interaction is only counted as having entered once relative to this STID assignment.
nAbandonedAcd	Integer	Int (4)	The number of ACD related interactions that abandoned this STID during the interval. Interactions that are reassigned from one STID to another STID (where no agent ever answers the interaction in the first STID) will not show up in *nAbandonedACD* or *nTransferACD*. They will only show in *nFlowOutAcd*. See the section in the Interval Queue Data summary section above for more information on abandons.
nGrabbedAcd	Integer	Int (4)	This value is not currently supported.

Column Name (Bold=Indexed)	**IC Data Type**	**SQL Data Type (Size)**	**Description**
nLocalDisconnectAcd	Integer	Int (4)	Number of ACD interactions that were disconnected locally. A local disconnect is defined as a disconnect by an IC agent/user or the IC system. A remote disconnect comes from the external party or phone company.
nAlertedAcd	Integer	Int (4)	Number of ACD related interactions that were in an “Alerting” state while in this STID (interactions which transition to “Alerting” more than once are counted once).
nAnsweredAcd	Integer	Int (4)	Number of ACD related interactions that were answered by agents while assigned to this STID. Answered interactions are interactions that reached a “Client_Connected” state with an agent. An interaction is only counted as answered once relative to this STID assignment. If the interaction is transferred to and/or answered by another agent or queue while still assigned to the same STID, it is still only counted as answered once for the STID. Voice mail interactions are considered abandons and do not count towards *nAnsweredAcd*. This value can exceed *nEnteredACD* for STID because interactions can enter the STID in the previous interval and be answered in the current interval.

Column Name (Bold=Indexed)	**IC Data Type**	**SQL Data Type (Size)**	**Description**
nAnswered	Integer	Int (4)	Number of all interactions (ACD and non-ACD) that were answered by the agents while assigned to this STID. Answered interactions are interactions that reached a “Client_Connected” state with an agent. See *nAnsweredAcd* for more information.
nAcdSvcLvl	Integer	Int (4)	Number of seconds in the first service level. For more information, see the description of service levels in the Interval Queue Data summary section above.
nAnsweredAcdSvcLvl1	Integer	Int (4)	Number of ACD interactions answered in first service level. The time tracked begins from the point the call becomes an ACD cal,; which is not necessarily the same as when it is assigned to the STID. For more information, see the description of service levels in the Interval Queue Data summary section above.
nAnsweredAcdSvcLvl2	Integer	Int (4)	Number of ACD interactions answered in second service level. See *nAnsweredAcdSvcLvl1*.
nAnsweredAcdSvcLvl3	Integer	Int (4)	Number of ACD interactions answered in third service level. See *nAnsweredAcdSvcLvl1*.
nAnsweredAcdSvcLvl4	Integer	Int (4)	Number of ACD interactions answered in fourth service level. See *nAnsweredAcdSvcLvl1*.
nAnsweredAcdSvcLvl5	Integer	Int (4)	Number of ACD interactions answered in fifth service level. See *nAnsweredAcdSvcLvl1*.
nAnsweredAcdSvcLvl6	Long	Int (4)	Number of ACD interactions answered in Sixth service level. See *nAnsweredAcdSvcLvl1*.

Column Name (Bold=Indexed)	IC Data Type	SQL Data Type (Size)	Description
nAbandonAcdSvcLvl1	Integer	Int (4)	Number of ACD interactions abandoned in first service level. The time tracked begins from the point the call becomes an ACD call; which is not necessarily the same as when it was assigned to the STID. For more information, see the description of service levels and abandoned calls in the Interval Queue Data summary section above.
nAbandonAcdSvcLvl2	Integer	Int (4)	Number of ACD interactions abandoned in second service level. See *nAbandonAcdSvcLvl1*.
nAbandonAcdSvcLvl3	Integer	Int (4)	Number of ACD interactions abandoned in third service level. See *nAbandonAcdSvcLvl1*.
nAbandonAcdSvcLvl4	Integer	Int (4)	Number of ACD interactions abandoned in fourth service level. See *nAbandonAcdSvcLvl1*.
nAbandonAcdSvcLvl5	Integer	Int (4)	Number of ACD interactions abandoned in fifth service level. See *nAbandonAcdSvcLvl1*.
nAbandonAcdSvcLvl6	Long	Int (4)	Number of ACD interactions abandoned in sixth service level. See *nAbandonAcdSvcLvl1*.
tGrabbedAcd	Integer	Int (4)	This value is not currently supported.
tAnsweredAcd	Integer	Int (4)	The sum of the time, in seconds, all ACD interactions assigned to the STID spent before entering “Client_Connected” state. The time tracked begins from the point the call becomes an ACD call; which is not necessarily the same as when it was assigned to the STID.
mtAnsweredAcd	Integer	Int (4)	The maximum time, in

Column Name (Bold=Indexed)	IC Data Type	SQL Data Type (Size)	Description
			seconds, an ACD interaction assigned to the STID was in the STID before entering a "Client_Connected" state. The time tracked begins from the point the call becomes an ACD call; which is not necessarily the same as when it was assigned to the STID.
tAbandonedAcd	Integer	Int (4)	The sum of the time, in seconds, all abandoned ACD interactions assigned to the STID were in the STID before they abandoned during this interval. The time tracked begins from the point the call becomes an ACD call, which is not necessarily the same as when it was assigned to the STID.
tTalkAcd	Integer	Int (4)	The sum of the time, in seconds, all ACD interactions assigned to the STID spent from when they first entered a "Client_Connected" state until the time the ACD interactions went inactive or flowed out of the STID. It is possible to have talk time appear in an interval without having any new ACD interactions for that interval. The time comes from ACD interactions that connected in that interval. Interactions might have been assigned to the STID in the previous interval. It is possible to have talk time on a chat. Think of *tTalkAcd* as the time an interaction is active with an agent
nHoldAcd	Integer	Int (4)	Number of ACD related interactions that were placed on hold while assigned to the STID.
tHoldAcd	Integer	Int (4)	The sum of the time, in seconds, all ACD interactions spent on hold while assigned

Column Name (Bold=Indexed)	**IC Data Type**	**SQL Data Type (Size)**	**Description**
			to this STID.
tAcw	Integer	Int (4)	The sum of the time, in seconds, agents spent in an “After Call Work” status (also known as wrap up time). The count starts when an interaction assigned to the STID goes inactive (usually due to a local or remote disconnect) and ends when the agent leaves the “After Contact Work” status. This value might or might not include non-ACD interactions depending on how the statuses are setup.
nExternToInternCalls	Integer	Int (4)	Number of interactions (ACD and non-ACD) originating from external locations and connecting to internal extensions.
nExternToInternAcdCalls	Integer	Int (4)	Number of ACD interactions from external locations to internal extensions.
nInternToExternCalls	Integer	Int (4)	Number of all interactions from internal extension to external locations.
nInternToExternAcdCalls	Integer	Int (4)	Number of ACD interactions from internal extension to external locations.
nInternToInternCalls	Integer	Int (4)	Number of all interactions from internal extensions to internal extensions. Internal to internal calls are counted as inbound calls whether an agent placed or received the call because IC has no way of determining the direction of internal calls.
nInternToInternAcdCalls	Integer	Int (4)	Number of ACD interactions from internal extensions to internal extensions. This includes ACD Interactions transferred between agents in the same STID.

Column Name (Bold=Indexed)	IC Data Type	SQL Data Type (Size)	Description
tExternToInternCalls	Integer	Int (4)	Sum of seconds for all interactions from external locations to internal extensions.
tExternToInternAcdCalls	Integer	Int (4)	Sum of seconds for ACD interactions from external locations to internal extensions.
tInternToExternCalls	Integer	Int (4)	Sum of seconds for all interactions from internal extensions to external locations.
tInternToExternAcdCalls	Integer	Int (4)	Sum of seconds for ACD interactions from internal extensions to external locations. (Not Implemented or used in this release, but it is set when using Interaction Dialer)
tInternToInternCalls	Integer	Int (4)	Sum of seconds for all interactions from internal extensions to internal extension.
tInternToInternAcdCalls	Integer	Int (4)	Sum of seconds for ACD interactions from internal extensions to internal extensions.
nAcwCalls	Integer	Int (4)	Number of outbound interactions made by agents during “After Contact Work” time. If an agent places an interaction after handling an ACD interaction assigned to the STID, that interaction, or any other interaction placed during the ACW time, is considered a *nACWcall*. The STID assignment is made on the initial ACD interaction and the ACW time is associated with that interaction and STID along with any outbound interaction made by the agent during the ACW time. nACWCall is also accumulated when an agent puts an ACD

Column Name (Bold=Indexed)	**IC Data Type**	**SQL Data Type (Size)**	**Description**
			interaction on hold and places a consult call with another party. IC assumes that any call placed with an ACD interaction on hold is nACWCall activity. IC actually caches the last set of containers associated with the agent's last ACD interaction. This cache is reset any time the agent answers another ACD interaction. When IC sees an ACD interaction or conditions on the agent that would count as an Acw interaction, it processes the time and the count against the cached container list. It is done this way because the ACD interaction that made this association might actually be dead and gone from the system by the time the *AcwCall* event occurs.
tAcwCalls	Integer	Int (4)	Sum of time, in seconds, agents spent on outbound interactions during “After Contact Work” time associated with an interaction that was assigned to the STID. Also see *nAcwCalls*.
nTransferedAcd	Integer	Int (4)	Number of ACD interactions transferred while assigned to this STID. The destination could have been another agent in the same distribution queue, or an agent outside of the distribution queue. It is not possible to transfer an interaction out of a STID in the usual sense. They must be removed and will then only show up in nFlowOutAcd. Only set for call interaction types.

Column Name (Bold=Indexed)	**IC Data Type**	**SQL Data Type (Size)**	**Description**
nNotAnsweredAcd	Integer	Int (4)	Number of ACD interactions that were not answered when presented to agents as an “Alerting” interaction.
tAlertedAcd	Integer	Int (4)	Sum of the time, in seconds, ACD interactions spent in an “Alerting” state on agent queues. Also referred to as ring time.
nFlowOutAcd	Integer	Int (4)	The number of ACD interactions that flowed out during this interval. Flow outs are defined as interactions that were removed from the STID without reaching an inactive state. Queue interactions have an inactive state that is used to mark the interaction for deletion by the ACD subsystem after the interaction has either been answered or abandoned. The most common reason for an interaction assigned to a STID to not reach an inactive state, and thus flow out, would be an interaction not answered or abandoned before the interaction is removed to the STID.
tFlowOutAcd	Integer	Int (4)	Sum of seconds ACD interactions were assigned to the STID before being counted in *nFlowOutAcd*. See *nFlowOutAcd* for more information. The time tracked begins from the point the interaction becomes an ACD interaction, which is not necessarily the same as when it is assigned to the STID.
nStartWaitAlertAcdCalls	Integer	Int (4)	Number of ACD interactions that were waiting to be answered or were alerting to be answered at the start of the interval.

Column Name (Bold=Indexed)	IC Data Type	SQL Data Type (Size)	Description
nStartActiveAcdCalls	Integer	Int (4)	Number of ACD interactions that were active with an agent at start of the interval.
nStartHeldAcdCalls	Integer	Int (4)	Number of ACD interactions that were held at start of the interval.
nEndWaitAlertAcdCalls	Integer	Int (4)	Number of ACD interactions that are waiting to be answered or were alerting to be answered at the end of the interval.
nEndActiveAcdCalls	Integer	Int (4)	Number of ACD interactions that are active with an agent at the end of the interval.
nEndHeldAcdCalls	Integer	Int (4)	Number of ACD interactions that are held at the end of the interval.
nTransferWithinAcdCalls	Integer	Int (4)	Not implemented in the current release. Number of ACD interactions transferred within this STID.
nTransferOutAcdCalls	Integer	Int (4)	Not implemented in the current release. Number of ACD interactions transferred out of this STID.
CustomValue1	Integer	Int (4)	Custom value for customer use.
CustomValue2	Integer	Int (4)	Custom value for customer use.
CustomValue3	Integer	Int (4)	Custom value for customer use.
CustomValue4	Integer	Int (4)	The default value is set to mtAbandonedACD. mtAbandonedACD is the maximum amount of time a customer waited before they abandoned in an interval. If a customer is using this custom value for their own purpose their customizations will override this default setting and the customizations will continue to work as configured.

Column Name (Bold=Indexed)	**IC Data Type**	**SQL Data Type (Size)**	**Description**
CustomValue5	Integer	Int (4)	The default value is set to nMessageACD. NMessageACD is the number of ACD interactions that went to a message state (voice mail for interactions) during the interval. It should be noted that subtracting this value from nAbandonedACD will not give you a true voice mail count, since it is possible for interactions to go in and out of voice mail several times or for an interaction to be answered and then sent to voice mail. If a customer is using this custom value for their own purpose their customizations will override this default setting and the customizations will continue to work as configured.
CustomValue6	Integer	Int (4)	The default value is set to nRequestedSuperAssistACD. NRequestSuperAssistACD is the number of supervisor-assist requests that were placed from the “clients” for ACD interaction during the interval. If a customer is using this custom value for their own purpose their customizations will override this default setting and the customizations will continue to work as configured.
I3TimeStampGMT	(System Supplied)	DateTime (8)	System supplied date time when the row was sent to the IC Logging server for insertion into the table.
SiteId	Integer	SmallInt (2)	Site identifier of the source of this interaction row. Used in multi-site rollup to distinguish the origin of data.
SubSiteId	Integer	SmallInt (2)	Not used in the current release. The purpose of this field is to divide use within a site, such as for multiple

Column Name (Bold=Indexed)	IC Data Type	SQL Data Type (Size)	Description
			independent companies on one IC system. Until IC has such functionality, this field is zero and is reserved for future use
cHKey3	String	Varchar (50)	This represents the third level of hierarchy grouping in terms of reporting. The most common value for this field will be the interaction type. This field will have a default value of "*". For more information, see "Record Type Hierarchy" earlier in this document.
cHKey4	String	Varchar (50)	This represents the fourth level of hierarchy grouping in terms of reporting. This field will not be used in a typical implementation. This field will have a default value of "*". For more information, see "Record Type Hierarchy" earlier in this document.

Index Definitions

Index Name	Index Type	Column Name	Order
I3TimeStampGMT	Clustered non-unique	I3TimeStampGMT	Ascending
		SiteId	Ascending
		SubSiteId	Ascending
PKQueueName	Primary Key	cName	Ascending
		cReportGroup	Ascending
		cHKey3	Ascending
		cHKey4	Ascending
		cType	Ascending
		I3TimeStampGMT	Ascending
		dIntervalStart	Ascending
		SiteId	Ascending
		SubSiteId	Ascending
IEIntervalStart	Non Unique	dIntervalStart	Ascending
		cName	Ascending

Index Name	Index Type	Column Name	Order
		SiteId	Ascending
		SubSiteId	Ascending

Interval Line and Line Group Data

IC generates Interval line and line group statistics for each line and reporting line group configured and stores the statistics in the *IlineStats* table and the *IlineGroupStats* table.

Data Source

The source of this data is Interval generated summary information about activity on lines and line groups.

Type of information

The types of information in the Interval Line and Line Groups Data is the length of interval, the sum of seconds the lines were active, the sum of seconds all lines were busy, the number of interactions, the total time for all traffic and outbound traffic (subtract outbound from all to get inbound), and the number of outbound interactions that were blocked from making a interaction.

Report Types using this data

The Line Group report types use line and line group information for both reports and charts.

Interval Line Statistics Table

This table logs interval statistics for individual lines you have configured in IC.

Physical Attributes

Log Identifier	Log Name	Table Name
82	Interval Line Statistics	IlineStats

Column Definitions

Column Name (Bold=Indexed)	IC Data Type	SQL Data Type (Size)	Description
LineId	String	Varchar (50)	Line name or identifier as configured in IC.
dIntervalStart	DateTime	DateTime (8)	Starting date/time of the interval. The interval is a server parameter value setup in IA. The default interval is 1800 seconds (30 minutes) and is always relative to the hour. In some cases, the interval start times might not be exactly on the hour. This is usually due to stopping and starting IC. These odd interval start times might be visible in the reports. See Appendix B

Column Name (Bold=Indexed)	**IC Data Type**	**SQL Data Type (Size)**	**Description**
			for more information.
nDuration	Integer	SmallInt (2)	Duration in seconds of the interval. If *dIntervalStart* is an odd start time, the duration of the interval will have a non standard length.
nEntered	Integer	SmallInt (2)	Number of interactions that entered or appeared on this line.
tSeized	Integer	Int (4)	Sum of time that the line was busy or seized for the interval.
nEnteredOutbound	Integer	SmallInt (2)	Number of interactions that entered the line for outbound use.
nOutboundBlocked	Integer	SmallInt (2)	Number of interactions that were blocked because the line was busy when they tried to enter the line for outbound use.
tResourcesAvailable	Integer	Int (4)	Sum of time the line was available for service. The system may take a line out of service because of detected signaling problems.
I3TimeStampGMT	(System Supplied)	DateTime (8)	System supplied date time when the row was sent to the IC Logging server for insertion into the table.
SiteId	Integer	SmallInt (2)	Site identifier of the source of this interaction row. Used in multi-site rollup to distinguish the origin of data.
SubSiteId	Integer	SmallInt (2)	Not used in the current version. The purpose of this field is to divide use within a site, such as for multiple independent companies on one IC system. Until IC has such functionality, this field is zero and is reserved for future use

Index Definitions

Index Name	Index Type	Column Name	Order
I3TimeStampGMT	Clustered non-unique	I3TimeStampGMT	Ascending
		SiteId	Ascending
		SubSiteId	Ascending
IEILineStats	Non-unique	dIntervalStart	Ascending
		SiteId	Ascending
		SubSiteId	Ascending
		LineId	Ascending

Interval Line Group Statistics Table

This table logs interval statistics for line groups you have configured for reporting in IC. Reporting line groups are line groups that have their *Reporting* attribute activated. This attribute is configured in Interaction Administrator. IC does not create line groups by default. The reporting line groups must be created before IC creates any meaningful data for the line group reports.

Note that the line group information is summary data about the lines in the group collected during the interval. Because of this, it is possible that someone could have added or removed lines from the configuration during the interval. This does not cause any computation problems, because IC does not record the number of lines, but instead records the total number of seconds the lines were in the configuration. If you divide this total by the seconds in the duration, you compute a fractional number of lines for this interval.

Physical Attributes

Log Identifier	**Log Name**	**Table Name**
81	Interval Line Group Statistics	IlineGroupStats

Column Definitions

Column Name (Bold=Indexed)	**IC Data Type**	**SQL Data Type (Size)**	**Description**
GroupId	String	Varchar (50)	Group Identifier as configured in IC. The line group must have the *ReportFlag* set true (in IA) for this data to be generated.
dIntervalStart	DateTime	DateTime	Starting date/time of the interval. The interval is a server parameter value setup in IA. The default interval is 1800 seconds (30 minutes) and is always relative to the hour. In some cases, the interval start times might not be exactly on the hour. This is usually due to stopping and starting IC. These odd interval start times might be visible in the reports. See Appendix B for more information.

Column Name (Bold=Indexed)	IC Data Type	SQL Data Type (Size)	Description
nDuration	Integer	SmallInt (2)	Duration in seconds of the interval. If *dIntervalStart* is an odd start time, the duration of the interval will have a non standard length.
nEntered	Integer	SmallInt (2)	Number of interactions that entered or appeared on lines in this group.
mEntered	Integer	SmallInt (2)	Maximum number of interactions that entered, or appeared at any point in time during this interval.
tActiveLines	Integer	Int (4)	The Sum of time, in seconds, each line in the group was active during this interval. For most intervals this would be *N* (for each line) * *nDuration*. Since lines might be activated or de-activated during the interval, this number might contain a portion of time that a line was active. *tActiveLines / nDuration* equals the average number of active lines that were in this line group for the interval. If there were no changes to the line group in the interval, the number will be whole.
tAllBusy	Integer	Int (4)	Sum of time, in seconds, ALL lines in this group were busy or seized during this interval.
tSeized	Integer	Int (4)	Sum of time, in seconds, lines in this group were busy or seized for interval.
nEnteredOutbound	Integer	SmallInt (2)	Number of interactions that entered the line group for outbound use.

Column Name (Bold=Indexed)	IC Data Type	SQL Data Type (Size)	Description
nOutboundBlocked	Integer	SmallInt (2)	Number of interactions that tried to enter the line group for outbound use, but were blocked because of ALL lines being busy.
tResourcesAvailable	Integer	Int (4)	Sum of time, in seconds, lines active in this group were available for service. This should not exceed *tActiveLines*, but can be less if the system takes a line out of service because of detected signaling problems.
I3TimeStampGMT	(System Supplied)	DateTime (8)	System supplied date time when the row was sent to the IC Logging server for insertion into the table.
SiteId	Integer	SmallInt (2)	Site identifier of the source of this interaction row. Used in multi-site rollup to distinguish the origin of data.
SubSiteId	Integer	SmallInt (2)	Not used in the current version. The purpose of this field is to divide use within a site, such as for multiple independent companies on one IC system. Until IC has such functionality, this field is zero and is reserved for future use.

Index Definitions

Index Name	Index Type	Column Name	Order
I3TimeStampGMT	Clustered non-unique	I3TimeStampGMT	Ascending
		SiteId	Ascending
		SubSiteId	Ascending
IEILineGroupStats	Non-unique	dIntervalStart	Ascending
		SiteId	Ascending
		SubSiteId	Ascending

Index Name	Index Type	Column Name	Order
		GroupId	Ascending

Fax Envelope History Table

The Fax Envelope History table records all fax attempts made in the IC system. The attempts can either be successful or fail to transmit the fax. Each fax is tracked by its envelope ID, call ID and fax ID. The envelope ID is an identifier for the person receiving the fax. It is possible to send the same fax to several different locations. The same fax to several different locations would result in several records with the same fax ID and different envelope IDs. Each time the system attempts to receive or send a fax, a call ID is created. The call ID recorded for each fax attempt will match the call IDs in the call detail log table.

The system will attempt to send faxes as long as the attempt count is less or equal to the retry count. When the attempt count becomes greater that the retry count, the system will no longer attempt to transmit the fax. Each fax will start with one attempt, plus the number of retries.

It should be noted that this log is automatically activated, but data will only be logged if there is a fax card set up in the system.

Physical Attributes

Log Identifier	83
Log Name	Fax Envelope History Log
Table Name	FaxEnvelopeHist

Column Definitions

Column Name (Bold = Indexed)	IC Data Type	Output Data Type (size)	Description
EnvelopeID	Long	Int	An identifier for the fax envelope. Each received or sent fax number will have its own envelope ID.
EnvelopeTimeStamp	Long	Int	A time stamp for the envelope ID. The EnvelopeTimeStamp is used with the Envelope ID to create a unique value over time. This timestamp cannot be translated into any usable real world date or time.
FaxID	Long	Int	An identifier for the fax. Each fax received or sent will have a fax ID. It is possible for the same fax to have multiple envelope IDs if sent to multiple recipients.

Column Name (Bold = Indexed)	**IC Data Type**	**Output Data Type (size)**	**Description**
FaxTimeStamp	Long	Int	A time stamp for the fax ID. The FaxTimeStamp is used with the Fax ID to create a unique value over time. This timestamp cannot be translated into any usable real world date or time.
CallIDKey	String	VarChar(18)	An identifier for the call ID used to receive or make the fax attempt. This identifier should be equal to the call ID in the CallDetail log
ProcessingDatetime	DateTime	DateTime(8)	The date and time that the fax was processed.
ProcessingDateTimeGMT	DateTime	DateTimeGMT	The date and time that the fax was processed adjusted to Greenwich Mean Time
SuccessFlag	String	Char(1)	Indicates whether the fax transmission was a success or a failure. **0** = failure and **1** = success.
RemoteCSID	String	VarChar(50)	The station ID of the remote fax machine. This is the fax machine that is external to the IC system.
RemoteNumber	String	Char(50)	The fax number of the remote fax machine. This is the fax machine that is external to the IC system.
T30	String	VarChar(50)	Fax specific addressing. This information will be specific to the fax machine and contains the T.30 string sent from the remote fax machine or sent by the IC system.
PortName	Sting	VarChar(50)	The IC port name where the fax was processed.
PortNumber	Long	Int	The IC port number where the fax was processed
Duration	Long	Int	The length of the transmission expressed in seconds
Speed	Long	Int	The speed of the transmission expressed in BPS
PageCount	Long	Int	The number of pages transmitted.

Column Name (Bold = Indexed)	**IC Data Type**	**Output Data Type (size)**	**Description**
ErrorInfo	String	VarChar(1024)	Internal IC errors on why faxes failed to transmit. This includes driver and TS errors.
SignalQuality	Long	Int	The quality of the fax transmission expressed as unit/range specific to the fax hardware used in the transmission and might not be provided with all fax hardware.
SignalStrength	Long	Int	The strength of the signal expressed as unit/range specific to the fax hardware used in the transmission and may not be provided with all fax hardware.
LineNoise	Long	Int	The amount of line noise expressed as unit/range specific to the fax hardware used in the transmission and might not be provided with all fax hardware.
Header	String	VarChar(128)	Transmitting Fax Only: The header information at the top of each fax.
SendWhen	Long	Int	Transmitting Fax Only: A flag indicating when the fax will be transmitted. **0** = ASAP, **1** = Scheduled Time, and **2** = Cheap hours.
CheapBeginDateTime	Datetime	Datetime	Transmitting Fax Only: The beginning of the off or cheap hours. This information is only present if SendWhen = **2**.
CheapEndDateTime	Datetime	Datetime	Transmitting Fax Only: The end of the off or cheap hours. This information is only present if SendWhen = **2**
ScheduledDateTime	Datetime	Datetime	Transmitting Fax Only: Scheduled time to send fax. This information is only present if SendWhen = **1**
Retries	Long	Int	Transmitting Fax Only: Number of retries requested for the fax. Each fax, by default, will have one attempt. The first attempt is not included in the retry number.

Column Name (Bold = Indexed)	**IC Data Type**	**Output Data Type (size)**	**Description**
RetryDelay	Long	Int	Transmitting Fax Only: The number of seconds between fax attempts
SubmitDateTime	DateTime	Datetime(8)	The date and time that the fax was submitted to the IC system for processing. The SubmitDateTime will match the ProcessingDatetime for incoming faxes.
SenderName	Sting	Char(50)	Transmitting Fax Only: The IC User ID of the user who submitted the fax for processing.
NotifyOnSuccess	Long	Int	Transmitting Fax Only: This value will be 1 if the sender is to be notified on successful fax transmission. By default the value is zero.
SuccessAddress	String	Char(100)	Transmitting Fax Only: If NotifyOnSuccess = 1, then this is the e-mail address of the person/persons to be notified of successful fax transmission.
NotifyOnFailure	Long	Int	Transmitting Fax Only: This value will be 1 if the sender is to be notified on failed fax transmissions. By default the value is zero.
FailureAddress	String	Char(100)	Transmitting Fax Only: If NotifyOnSuccess = 1, then this is the email address of the person/persons to be notified of failed fax transmissions.
FailureAttempts	Long	Int	Transmitting Fax Only: The number of the fax attempt. When the number of FailureAttempts is greater than the Retries, the fax is considered to be failed.
FailureType	Long	Int	Transmitting Fax Only: The reason the fax failed to transmit. **1** = Busy, **2** = No Answer, **3** = No Remote Fax Detected, and **4** = Unknown. **0** = Successful Transmission

Column Name (Bold = Indexed)	IC Data Type	Output Data Type (size)	Description
MaxBPS	Long	Int	Transmitting Fax Only: The maximum allowed BPS. This value can be set by the sender.
DeviceGroup	String	Char(50)	Transmitting Fax Only: The user requested device group to be used to transmit the fax. In IA, you can create named groups of fax stations. For example, imagine this scenario that has 4 fax stations, which are used for two purposes: First to handle normal inbound/outbound daily faxing, and second to handle large fax broadcasts to thousands of users. Without fax groups, a fax broadcast to thousands of users would tie up all the fax stations for hours or days and your normal inbound/outbound faxing would be down. In this case you could, for example, create a fax group called FaxBroadcast which includes only two of the fax stations and set this fax group name on each broadcast fax. This will restrict the faxes to using only stations included in the group.
CoverPageName	String	VarChar(50)	Transmitting Fax Only: The name of the type of cover page used, such as **Confidential**, **Fax1**, **Fax2**
ToCompany	String	Char(50)	Transmitting Fax Only: The name of the company of the fax recipient as entered on the cover page.
ToName	String	Char(50)	Transmitting Fax Only: The name of the fax recipient as entered on the cover page.
ToVoicePhone	Sting	VarChar(50)	Transmitting Fax Only: The telephone number of the fax recipient as entered on the cover page.

Column Name (Bold = Indexed)	**IC Data Type**	**Output Data Type (size)**	**Description**
FromName	String	Char(50)	Transmitting Fax Only: The name of the fax sender as entered on the cover page. This name can be different from the SenderName as this name is not system driven, but entered by the sender.
FromFaxPhone	String	VarChar(50)	Transmitting Fax Only: The fax number of the sender as entered on the cover page.
FromVoicePhone	String	VarChar(50)	Transmitting Fax Only: The phone number of the sender as entered on the cover page.
FromCompany	String	Char(50)	Transmitting Fax Only: The name of the sending company as entered by the sender on the cover page.
FaxComment	String	VarChar(1024)	Transmitting Fax Only: Any comments entered by the sender for the fax.
LocalCSID	String	VarChar(50)	The station ID of the internal (IC) fax machine. This is the fax machine that is internal to the IC system.
I3TimeStampGMT	System Supplied	DateTimeGMT	The system supplied date and time of when the row was added or inserted into the table.
SiteID	Long	Int	Site identifier of the source for this record. Used in multi-site installations to distinguish the origin of the record. Configured when setting up IC.
SubSiteID	Long	Int	Not used in this release. The purpose of this field is to divide use within a site, such as for multiple independent companies on one IC system. Until IC has such functionality, this field is zero and is reserved for future use.
Direction	Long	Int	The direction of the fax. **1** = incoming and **0** = outgoing.
SubmitDateTimeGMT	Datetime	DatetimeGMT	The Greenwich Mean Time of SubmitDateTime

Index Definitions

Index Name	Index Type	Column Name	Order
I3TimeStampGMT I3TimeStampGMT83 (Oracle)	Clustered non-unique Non-unique	I3TimeStampGMT	Ascending
		SiteID	Ascending
		SubSiteID	Ascending
FaxEnvelopeHist	Unique	EnvelopeID	Ascending
		EnvelopeTimestamp	Ascending
		CallIDKey	Ascending
		SiteID	Ascending
		SubSiteID	Ascending
FaxEnvelopeHistProcTime	Non-Unique	ProcessingDateTime	Ascending
		SiteID	Ascending
		SubSiteID	Ascending
FaxEnvelopeHistCallID	Non-unique	CallIDKey	Ascending
		SiteID	Ascending
		SubsiteID	Ascending
FaxEnvelopeFaxID	Non-unique	FaxID	Ascending
		FaxTimestamp	Ascending
		SiteID	Ascending
		SubSiteID	Ascending

IVR History Table

This log contains historical data about activity within the IVR.

Physical Attributes

Log Identifier	84
Log Name	IVR History
Table Name	IVRHistory

Column Definitions

Column Name	**Type**	**Description**
InteractionKey	String(10)	Unique string identifier for an Interaction. Same as CallId in CallDetail table, or CallIdKey in FaxEnvelopeHist table.
SiteId	Integer	Site identifier for the interaction
SubSiteId	Integer	Reserved.
dEventTime	Datetime	Datetime event occurred
SeqNo	Integer	Sequence number of event. If more than one event occurs for this InteractionKey in a second, then the sequence number will be incremented for each additional interaction that occurred. Normally this value will be 0.
cEventType	String(10)	Event that has occurred. The event will define the meaning of the columns that follow. 'INAPP' - Enter application (ApplicationName) 'OUTAPP' - Exit application (ApplicationName, AbortFlag) 'INMENU' – Enter menu (MenuName) 'OUTMENU' – Exit menu (MenuName, AbortFlag) 'PATH' – Path taken (Path Name, AbortFlag) includes agent and disconnect.
cEventData1	String(50)	See cEventType for definition of this value.
cEventData2	String(50)	See cEventType for definition of this value.
cEventData3	String(50)	See cEventType for definition of this value.
I3TimestampGMT	Datetime	Timestamp value created at time of data creation used in automated data deletion process.

IVR Interval Table

This log contains IVR navigation data that is accumulated and reported on a configurable interval.

Physical Attributes

Log Identifier	85
Log Name	IVR Interval
Table Name	IVRInterval

Column Definitions

Name	Type	Description
cLevelName	String(50)	Application for which this data is reported. "*" will be used when reporting for all activities in IVR. This will be the concatenation of the Level Names defining the current level separated by '|'. Example 'Profile1|Schedule2|Application3'
nLevel	Tinyint	Level of the statistics. Profile=1, Schedule=2, Application/Complex Operation=3, Menu=4, Task=5.
ParentLevels	String(200)	
cExitPath	String(250)	Path for which this data is reported. An "*" will be used when this is a summary row of all activity for the Application and Menu. The path reported is the exit path from the menu. When the exit reason is transfer to a Queue, then the path will be reported as "Transfer WG:{Queue Name}". When the exit reason is a transfer to a user or station queue, the path will be "Transfer User" or "Transfer Station". When the exit reason is an abort of IVR, then the path will be "Abort". "Disconnect" will be used when an interaction disconnects while still in IVR.
SiteId	Integer	Site Identifier of site reporting data.
SubSiteId	Integer	Reserved for future use
dIntervalStart	Datetime	Starting time of the interval.
dIntervalStartGMT	Datetime	Starting time of the interval adjusted to GMT.
nDuration	Integer	Duration of reporting interval in seconds starting from the dIntervalStart.
nEnteredFirst	Integer	Number of interactions entering this level/ path the first time.

Name	Type	Description
nEnteredRepeat	Integer	Number of interactions repeating an entry to this level / path.
nDurationFirst	Integer	Seconds from entry until exit of level taking path for the first time.
nDurationRepeat	Integer	Seconds from reentry until exit of level or taking path again.
nExitCode	Integer	Code indicating the type of exit associated with the level0 = Not an abort, 1 = Abort. This might contain other values in the future as other exit types are added.
I3TimestampGMT	Datetime	Timestamp value created at time of data creation used in automated data deletion process.

WrapUp Statistics

If a wrap-up code is used on a call, that code will be added to the Call Detail Record log entry for that call. Wrap-up code statistics are stored in a separate wrap-up codes table, report log 93. The wrap-up code statistics are collected on the same interval as the other queue statistics. The data follows the same hierarchy as the current Agent hierarchy. If multiple agents enter a wrap-up code for the same interaction, only the last agent's entry is stored in the table. This data is logged in addition to the regular interval statistics when wrap up codes are activated for a queue. The hierarchy is Agent, Queue, Wrap Up Code and Media Type.

Wrap-up code reports are similar to other call detail and summary reports except that they are organized or sorted by the wrap-up code names available to an agent or workgroup.

Physical Attributes

Log Identifier	93
Log Name	WrapUp Statistics
Table Name	IWrapUpStats

Column Definitions

Column Name **Bold = Indexed**	**IC Data Type**	**SQL Data Type (Size)**	**Description**
cName	String	Varchar2 (50)	Name of the Agent (User) queue that originated these statistics.
cReportGroup	String	Varchar2 (50)	Subgroup for *cName*. Agent (User) queues automatically have a report group for each workgroup that they are a member of. A special '*' report group represents the sum of all activity for the *cName* in this interval. Any custom reports groups that might be assigned to interactions that the agent handles are also automatically added to the records associated with the *cName.*
cHKey3	String	Varchar2 (50)	This represents the third level of hierarchy grouping in terms of reporting. The most common value for this field will be the interaction type. This field will have a default value of "*". For more information, see "Record Type Hierarchy" earlier in this document.

Column Name **Bold = Indexed**	**IC Data Type**	**SQL Data Type (Size)**	**Description**
cHKey4	String	Varchar2 (50)	This represents the fourth level of hierarchy grouping in terms of reporting. This field will not be used in a typical implementation. This field will have a default value of "*". For more information, see the record type hierarchy section in the Interval Queue Data summary.
cType	String	Char (1)	Type of Queue statistics data. In the case of the Agent Queue Statistics Interval table, an "**A**" will be entered to signify the type is Agent (User).
dIntervalStart	DateTime	DateTime (8)	Starting date/time of the interval. The interval is a server parameter value setup in IA. The default interval is 1800 seconds (30 minutes) and is always relative to the hour. In some cases, the interval start times might not be exactly on the hour. This is usually due to stopping and starting IC. These odd interval start times might be visible in the reports. See Appendix B for more information.
nDuration	Integer	Int (4)	The duration in seconds of the interval. If *dIntervalStart* is an odd start time, the duration of the interval will have a non standard length.
nCompleted	Integer	Int (4)	Number of ACD related queue items that were answered by the agent. Answered interactions are interactions that reached a “Client_Connected” state with an agent. This number can exceed *nEnteredACD* for the agent’s queue because interactions can enter the queue in the previous interval and then be answered in the current interval. Items are only answered once relative to the agent queue assignment.

Column Name **Bold = Indexed**	**IC Data Type**	**SQL Data Type (Size)**	**Description**
tTalked	Integer	Int (4)	The sum of the time, in seconds, all ACD items spent from when they first entered a "Client_Connected" state until the time the ACD items went inactive or flowed out of the queue. It is possible to have talk time appear in an interval without having any new ACD items for that interval. The time comes from ACD items that connected in that interval. Items might have entered the queue in the previous interval. It is possible to have talk time on a chat. Think of *tTalkAcd* as the time an interaction is active with an agent.
tAcw	Integer	Int (4)	The sum of the time, in seconds, the agent spent in an "After Contact Work" status (also known as wrap up time). The count starts when an item goes inactive, usually due to a local or remote disconnect, and the count ends when the agent leaves the "After Contact Work" status. This value might or might not include non-ACD items depending on how the statuses are setup.
nHold	Integer	Int (4)	Number of ACD items that were placed on hold while in this queue.
tHold	Integer	Int (4)	The sum of the time, in seconds, all ACD items spent on hold while in this queue.
I3TimeStampGMT	(System Supplied)	DateTime (8)	System supplied date time when the row was sent to the IC Logging server for insertion into the table.
SiteId	Integer	Smallint (2)	Site identifier of the source for this record. Used in multi-site installations to distinguish the origin of the record. Configured when setting up IC.
SubSiteId	Integer	Smallint (2)	The purpose of this field is to divide use within a site, such as for multiple independent companies on one IC system. Until IC has such functionality, this field is zero and is reserved for future use.
nSupervisorRequests	Integer	Int (4)	The number of ACD interactions where the agent requested supervisor assistance using the supervisor request feature in the client.

Agent Queue Activation History

This log contains data about each agent's activation and deactivation change history in each queue. It records an entry each time the agent is activated or deactivated from a workgroup with a queue, where the agent is a member of that workgroup.

Queue Activation

IC administrators, supervisors, and users with the appropriate administrative access controls can optionally activate and deactivate agents on a per queue basis without regard to the agent's Interaction Center status or state. See "Agent Status Data" earlier in this document for more information on agent status and state. This feature enables authorized agents monitoring multiple queues to deactivate themselves from inactive or lower priority queues in order to monitor busy or high priority queues without changing their status or logging out of the inactive queue. It also enables supervisors and administrators to activate or deactivate other agents, using Interaction Supervisor and Interaction Administrator, in queues without regard to that agent's status or logged in state. This only works on ACD and Custom workgroup queues.

The activation and deactivation event criteria includes:

- When an agent activates or deactivates himself or herself via the Workgroup Activation dialog in Interaction Client. This requires that the agent's user account be given "Activate Self" Access Control in Interaction Administrator.
- When a supervisor activates or deactivates an agent using Interaction Supervisor. This requires that the supervisor's user account be given "Activate Others" Access Control in Interaction Administrator.
- When a user is added or removed from an ACD or Custom workgroup in Interaction Administrator. By default, users are added to workgroups in an Activated state.
- When an ACD or Custom workgroup is created or deleted in Interaction Administrator. By default, workgroups are created with users in an Activated state.

All ACD agent user accounts that are members of ACD or Custom workgroup queues and which are upgraded from a pre-IC 2.4 system are flagged as Activated when they are imported into IC 2.4.

Physical Attributes

Log Identifier	86
Log Name	Agent Queue Activation History
Table Name	AgentQueueActivationHist

Column Definitions

Name	Type	Description
UserId	String(50)	The user name defined in Interaction Administrator for each agent logged into the system and who is a member of an ACD or Custom workgroup queue.
ActivationDateTime	DateTime	The local date and time an agent was activated or deactivated in a queue. By default, agents are activated or deactivated in the queue in the following conditions: 1) When an agent is activated or deactivated from the Workgroup Activation dialog in Interaction Client or Interaction Supervisor. 2) When a user is added or removed from an ACD or Custom workgroup in Interaction Administrator. 3) When an ACD or Custom workgroup is created or deleted in Interaction Administrator.
ActivationDateTimeGMT	DateTime	Greenwich Mean Time-based agent activation or deactivation date and time.
Workgroup	String(50)	The name of each ACD or Custom workgroup queue included in the parameter range.
ActivationFlag	Integer	When an agent is activated in a workgroup queue, the value is **1**, which appears as **Yes** in the report. When an agent is deactivated, the value is **0**, which appears as **No** in the Activated column of the report.
HasQueueFlag	Integer	Indicates if the named workgroup is defined in Interaction Administrator as a workgroup with a queue. In the current release, only workgroups with a queue can activate and deactivate agents. The value is **1** if the workgroup has a queue, which appears as **Yes** in the **Has Queue?** column. Otherwise, the value is **0**.
ActivatedBy	String(50)	Contains the IC user name of the person with Access Control who activated or deactivated the User ID from a queue in Interaction Client or Interaction Supervisor. If the change is made from Interaction Administrator, the value is "Administrator" followed by “User Added” or “User Removed”, or “Workgroup Added” or “Workgroup Removed”.

Name	Type	Description
SiteId	Integer	Site identifier of the source for this record. Used in multi-site installations to distinguish the origin of the record. Configured when setting up IC.
SubSiteId	Integer	Reserved for future use
I3TimestampGMT	Datetime	Timestamp value created at time of data creation used in automated data deletion process.

Administrative Data

This class of data includes the Interaction Administrator (IA) Change Notification table and the IC Change Notification table. These tables are audit trail tables for configuration changes you make in IC. Both of these tables record changes you make in IC, but they do so from different levels of the system.

Data Source

The data source for this class of data is event driven by changes you make in the configuration in Interaction Administrator, or any change you make to the IC configuration in the case of the IC Change Notification log.

Type of Information Stored

The type of information we store in this table are class, path, key, user and station making the change (not available for IC Change notification), and type of change.

Report Types using this data

The administrative report types use this data to allow you to search for changes by class and other attributes.

Interaction Administrator Change Notification Table

This table is a log of all changes you make using the Interaction Administrator program. IC adds a record in the IA Change Notification table for any change you make using Interaction Administrator. A change you make using handlers or other methods is not recorded here.

Physical Attributes

Log Identifier	**Log Name**	**Table Name**
7	Interaction Administrator Change Notification Log	IAChangeLog

Column Definitions

Column Name (Bold=Indexed)	**IC Data Type**	**SQL Data Type (Size)**	**Description**
ChangeDateTime	DateTime	DateTime (8)	Date/time of the change.
ChangeTime	String	Char (8)	Time of the change.
ChangeDateTimeGMT	DateTime	DateTime (8)	Date/time of the change adjusted to Greenwich Mean Time.

Column Name (Bold=Indexed)	**IC Data Type**	**SQL Data Type (Size)**	**Description**
UserId	String	Varchar (50)	User identifier of the user whose configuration has changed. This is the User ID as defined in Interaction Administrator
StationId	String	Varchar (50)	Identifier of station whose configuration has changed.
ChangeType	String	Varchar (50)	Type of change. Expected values are “Created”, “Deleted”, “Renamed”, “Modified”
EntryKey	String	Varchar (128))	Entry path of the directory services key changed. Includes key name.
EntryClass	String	Varchar (50)	Directory services class of the key that was changed. Typical values are *“User”* or *“Report Log”*.
I3TimeStampGMT	(System Supplied)	DateTime (8)	System supplied date/time when the row was sent to the IC Logging server for insertion into the database.
SiteId	Integer	SmallInt (2)	Site identifier of the source of this interaction row. Used in multi-site rollup to distinguish the origin of data.
SubSiteId	Integer	SmallInt (2)	Not used in IC 2.1. The purpose of this field is to divide use within a site, such as for multiple independent companies on one IC system. Until IC has such functionality, this field is zero and is reserved for future use

Index Definitions

Index Name	Index Type	Column Name	Order
I3TimeStampGMT	Clustered non-unique	I3TimeStampGMT	Ascending
		SiteId	Ascending
		SubSiteId	Ascending
IEChangeTime	Non Unique	ChangeDateTime	Ascending
		EntryClass	Ascending
		SiteId	Ascending
		SubSiteId	Ascending
IEEntryKey	Non Unique	EntryKey	Ascending
		ChangeDateTime	Ascending
		SiteId	Ascending
		SubSiteId	Ascending
IEUserId	Non Unique	UserId	Ascending
		ChangeDateTime	Ascending
		SiteId	Ascending
		SubSiteId	Ascending
IEStationId	Non Unique	StationId	Ascending
		ChangeDateTime	Ascending
		SiteId	Ascending
		SubSiteId	Ascending

IC Change Notification Table

The IC Change Notification table records any changes made through the Directory Services subsystem, including any configuration change. This includes every persistent user/agent status change made by a user or handler.

Note

These changes include security changes made to each user if the default user security is changed, so the number of entries this log generates could be very large, if you have it activated.

For this reason, IC configures this log and table inactive. If tighter auditing is required, you can activate the log to provide the information detailed below, at the cost of additional database space and system performance.

Physical Attributes

Log Identifier	**Log Name**	**Table Name**
8	IC Change Notification Log	ICDirChangeLog

Column Definitions

Column Name (Bold=Indexed)	**IC Data Type**	**SQL Data Type (Size)**	**Description**
ChangeDateTime	DateTime	DateTime (8)	Date/time of the change.
ChangeTime	String	Char (8)	Time of the change as a string.
ChangeDateTimeGMT	DateTime	DateTime (8)	Date/time of the change adjusted to Greenwich Mean Time.
NotificationType	String	Varchar (50)	Type of notification. Expected values are: • Created • Deleted • Renamed • Modified • Inactive Created • Inactive Deleted • Inactive Renamed • Inactive Modified Inactive types designate changes that occur in inactive portions of the configuration tree.
EntryName	String	Varchar (50)	Name of the entry changed.
EntryPath	String	Varchar (128)	Full path, including key, of the entry that was changed.

Column Name (Bold=Indexed)	**IC Data Type**	**SQL Data Type (Size)**	**Description**
EntryClass	String	Varchar (50)	Directory Services class of the key that was changed. Typical values are **User** or **Report Log**.
ListOfAttributes	String	Varchar (255)	List of the attributes that were modified or created.
I3TimeStampGMT	(System Supplied)	DateTime (8)	System supplied date time when the row was sent to the IC Logging server for insertion into the table.
SiteId	Integer	Long	Site identifier of the source of this interaction row. Used in multi-site rollup to distinguish the origin of data.
SubSiteId	Integer	Long	Not used in this release. The purpose of this field is to divide use within a site, such as for multiple independent companies on one IC system. Until IC has such functionality, this field is zero and is reserved for future use.

Index Definitions

Index Name	Index Type	Column Name	Order
I3TimeStampGMT	Clustered non-unique	I3TimeStampGMT	Ascending
		SiteId	Ascending
		SubSiteId	Ascending
IEChangeTime	Non Unique	ChangeDateTime	Ascending
		EntryClass	Ascending
		SiteId	Ascending
		SubSiteId	Ascending
IEEntryPath	Non Unique	EntryPath	Ascending
		ChangeDateTime	Ascending
		SiteId	Ascending
		SubSiteId	Ascending
IEEntryName	Non Unique	EntryName	Ascending
		ChangeDateTime	Ascending
		SiteId	Ascending
		SubSiteId	Ascending

Configuration Mirror Data

Configuration mirror data is data that IC copies from the configuration to aid in the generation of reports. The mirror tables include data for account code configurations, user and workgroup relationships, line and line group configurations, and line group to lines relationships.

The User to Workgroup relationship allows you to build reports that can group user data by workgroups. The line and line group reports use the line and line group configuration information to report on least used lines by group, or to display additional configuration information about the line or group. The account code configuration allows the system to display account code information on various interaction detail type reports.

Note that these tables reflect the current configuration. Reports that use these tables might not represent historical information correctly if you have changed the configuration information since IC generated the historical data. Currently, the reports that operate with this restriction are the Queue Period Statistics Agent Performance and Line Group reports.

Normally you would not use the logging tool to place data into these tables. IC updates these tables automatically via a system process. If this data is out of sync with IC, you can update the data for a site by executing the following command at the IC server command prompt:

```
″SendCustomNotification AdminServer SyncAllMirrorLogs″
```

Formulas in IC reports allow the account code information and the user workgroup relationship information to be read directly from Interaction Administrator. This allows for more functionality on the reports. The reports use the ININ function to get the requested information.

Data Source

The source for this data is the IC configuration which the system copies from IC to your database on any restart, and updated any time you change the configuration.

Types of information

Types of information include portions of configuration information for user to workgroup relationships, line and line group configuration, and line to line group relationships.

Report types using this data

Some queue period statistics reports use this data to restrict users to non-queue workgroups. Line group reports use this data to show details about lines and line group configurations, as well as to find the least used lines in a group.

Mirror Data and the Switchover Process

When you configure two IC servers so that one is an automatic backup of the other using the switchover process, no duplication of data or any problems relative to any report and configuration mirror data logging in IC should occur. This is because the system that switchover considers the backup operates in a suspended mode that suppresses the normal mirror and data logging activities.

However, the system might duplicate data if you disconnect one of the two systems from the switchover hardware and boot it up as a stand-alone server. This system then becomes a full and complete IC system that is identically configured to the other running server. This means that any operations that you perform on this *separated backup* IC system, will cause active mirror and logging activity to go to the shared database. This can result in IC inserting duplicate rows of data, or possibly, inserting the wrong data into the tables.

Caution

When you are using switchover, make sure that if you perform any tests while the IC systems are not in their normal switchover configuration, change the server Site ID to make sure there is no conflict with data being generated by the two separated IC systems.

User to Workgroup Relationship Mirror Table

The content of this table is a mirror image of users and their workgroup membership as you have them *currently* configured in IC. This table is not used on any of the IC reports. There are plans to remove this table from the database in a future release.

Physical Attributes

Log Identifier	**Log Name**	**Table Name**
70	User to Workgroup Relationship	UserWorkgroups

Column Definitions

Column Name (Bold=Indexed)	**IC Data Type**	**SQL Data Type (Size)**	**Description**
UserId	String	Varchar (50)	User identifier for a user configured in IC.
WorkGroup	String	Varchar (50)	Workgroup for which the user is currently a member in IC.
QueueFlag	Sting	Varchar (50)	Flag that indicates if the workgroup has a queue. Workgroups can be created without queues and are typically used for organization purposes at a site. This flag allows reports to use queue or non-queue workgroups, where appropriate, based on the type of report being executed. '**N**' = No queue, '**Y**' = Has Queue.
I3TimeStampGMT	(System Supplied)	DateTime (8)	System supplied date time when the row was sent to the IC Logging server for insertion into the table.

Column Name (Bold=Indexed)	IC Data Type	SQL Data Type (Size)	Description
SiteId	Integer	SmallInt (2)	Site identifier of the source of this interaction row. Used in multi-site rollup to distinguish the origin of data.
SubSiteId	Integer	SmallInt (2)	Not used in the current release. The purpose of this field is to divide use within a site, such as for multiple independent companies on one IC system. Until IC has such functionality, this field is zero and is reserved for future use

Index Definitions

Index Name	Index Type	Column Name	Order
I3TimeStampGMT	Clustered non-unique	I3TimeStampGMT	Ascending
		SiteId	Ascending
		SubSiteId	Ascending
PKWorkgroup	Primary Key	Workgroup	Ascending
		UserId	Ascending
		SiteId	Ascending
		SubSiteId	Ascending
IEUserId	Unique	UserId	Ascending
		Workgroup	Ascending
		SiteId	Ascending
		SubSiteId	Ascending

Line Configuration Mirror Table

The content of this table is a mirror image of the lines as you have them *currently* configured in IC.

Physical Attributes

Log Identifier	**Log Name**	**Table Name**
71	Line Configuration Mirror	LineConfig

Column Definitions

Column Name (Bold=Indexed)	**IC Data Type**	**SQL Data Type (Size)**	**Description**
LineId	String	Varchar (50)	Identifier for the line.
ActiveFlag	Integer	SmallInt (2)	Flag indicating if the line configuration is an active or inactive configuration. Normally, inactive lines would be ignored by reports. 1 = Active, 0 = Inactive.
Direction	String	Varchar (20)	Line direction.
LineType	String	Varchar (50)	Line or board type of the line. Expected values would be *Analog*, *T1*, and *E1*.
PhoneNumber	String	Varchar (20)	Phone number of the line.
I3TimeStampGMT	(System Supplied)	DateTime (8)	System supplied date/time when this row was sent to the IC Logging server for insertion into the table.
SiteId	Integer	SmallInt (2)	Site identifier of the source of this interaction row. Used in multi-site rollup to distinguish the origin of data.
SubSiteId	Integer	SmallInt (2)	Not used in this current release. The purpose of this field is to divide use within a site, such as for multiple independent companies on one IC system. Until IC has such functionality, this field is zero and is reserved for future use

Index Definitions

Index Name	Index Type	Column Name	Order
I3TimeStampGMT	Clustered non-unique	I3TimeStampGMT	Ascending
		SiteId	Ascending
		SubSiteId	Ascending
IELineConfig	Non-unique	LineId	Ascending
		SiteId	Ascending
		SubSiteId	Ascending

Line Group Configuration Mirror Table

The content of this table is a mirror image of the line groups as you have them *currently* configured in IC.

Physical Attributes

Log Identifier	Log Name	Table Name
72	Line Group Configuration Mirror	LineGroupConfig

Column Definitions

Column Name (Bold=Indexed)	IC Data Type	SQL Data Type (Size)	Description
GroupId	String	Varchar (50)	Group identifier or name of the line group.
Description	String	Varchar (50)	Description of the line group from the IC configuration.
DialGroupFlag	Sting	SmallInt (2)	Flag that indicates if the line group is used by IC for dialing. Not all groups need be configured for use as a dialing group. **1** = True, **0** = False.
ReportFlag	Sting	SmallInt (2)	Flag that indicates if the group is to be included in reports. If the flag as false, this line group will be excluded from line group reports. **1** = True, **0** = False.
I3TimeStampGMT	(System Supplied)	DateTime (8)	System supplied date/time when the row was sent to the IC Logging server for insertion into the table.
SiteId	Integer	SmallInt (2)	Site identifier of the source of this interaction row. Used in multi-site rollup to distinguish the origin of data.
SubSiteId	Integer	SmallInt (2)	Not used in this release. The purpose of this field is to divide use within a site, such as for multiple independent companies on one IC system. Until IC has such functionality, this field is zero and is reserved for future use.

Index Definitions

Index Name	Index Type	Column Name	Order
I3TimeStampGMT	Clustered non-unique	I3TimeStampGMT	Ascending
		SiteId	Ascending
		SubSiteId	Ascending
IELineGroupConfig		GroupId	Ascending
		SiteId	Ascending
		SubSiteId	Ascending

Line Group to Line Mirror Table

The content of this table is a mirror image of the line group to lines relationship as they are currently configured in IC.

Physical Attributes

Log Identifier	**Log Name**	**Table Name**
73	Line Group to Lines Relationship Mirror	LineGroupLines

Column Definitions

Column Name (Bold=Indexed)	**IC Data Type**	**SQL Data Type (Size)**	**Description**
GroupId	String	Varchar (50)	Group identifier for the line group.
LineId	String	Varchar (50)	Line identifier for a line in the Line Group.
I3TimeStampGMT	(System Supplied)	DateTime (8)	System supplied date/time when row was sent to the IC Logging server for insertion into the table.
SiteId	Integer	SmallInt (2)	Site identifier of the source of this interaction row. Used in multi-site rollup to distinguish the origin of data.
SubSiteId	Integer	SmallInt (2)	Not used in IC 2.1. The purpose of this field is to divide use within a site, such as for multiple independent companies on one IC system. Until IC has such functionality, this field is zero and is reserved for future use.

Index Definitions

Index Name	Index Type	Column Name	Order
I3TimeStampGMT	Clustered non-unique	I3TimeStampGMT	Ascending
		SiteId	Ascending
		SubSiteId	Ascending
IELineGroupLines	Non-clustered	GroupId	Ascending
		LineId	Ascending
		SiteId	Ascending
		SubSiteId	Ascending

Account Code Mirror

The content of this table is a mirror image of the account codes as they are currently configured in IC. This table is not used on any of the IC reports. There are plans to remove this table from the database in a future release.

Physical Attributes

Log Identifier	**Log Name**	**Table Name**
74	Account Code Mirror	AccountCodeMirror

Column Definitions

Column Name (Bold=Indexed)	**IC Data Type**	**SQL Data Type (Size)**	**Description**
AccountCode	String	Varchar (50)	Group identifier for the line group.
Description	String	Varchar (128)	Line identifier for a line in the Line Group.
I3TimeStampGMT	(System Supplied)	DateTime (8)	System supplied date/time when row was sent to the IC Logging server for insertion into the table.
SiteId	Integer	SmallInt (2)	Site identifier of the source of this interaction row. Used in multi-site rollup to distinguish the origin of data.
SubSiteId	Integer	SmallInt (2)	Not used in the current release. The purpose of this field is to divide use within a site, such as for multiple independent companies on one IC system. Until IC has such functionality, this field is zero and is reserved for future use.

Index Definitions

Index Name	Index Type	Column Name	Order
I3TimeStampGMT	Clustered non-unique	I3TimeStampGMT	Ascending
		SiteId	Ascending
		SubSiteId	Ascending
PKAccountCodeMirror	Non-clustered	AccountCode	Ascending
		SiteId	Ascending
		SubSiteId	Ascending

Appendix A: How IC generates Interaction IDs

This section describes how IC generates unique IDs for each interaction in the system. These identifiers are called **Call ID** in Interaction Client, and they are modified and stored in the IC database as unique **CallIDKey** in the **Call ID** column. However, these identifiers apply to all interaction types.

Call ID Keys changed in 2.3

The CallIDKey is comprised of the 10 digit CallId plus an eight digit date in this format: YYYYMMDD. For example, a call placed on April 9, 2004 might show a Call ID of 2101990183 on Interaction Client. The resulting CallIDKey stored in the database would be that CallID plus the date: 2101990183**20040409**.

Note

In IC version 2.2, the Call ID Key remained 10 characters but it became a *case-insensitive* string. In 2.3, it changed from 10 characters (a CHAR10) to 18 characters (CHAR18).

Prior to the release of version 2.3, the CallID and the CallIDKey were not the same, so you could not easily map these values to each other without customizations. Starting in 2.3, the CallIDKey included the CallID embedded within it. That meant, for the first time, you were able to search the Call Detail Record log for the original CallID number as seen in Interaction Client.

Call IDs and Call ID Keys changed in 2.1

In 2.1 the Call ID, as seen in the client, changed to a single digit (1-3), followed by the Site ID, then followed by a six-digit sequence number.

The Client Call ID: 1001010003

The Call ID no longer "resets" at midnight. The number is persisted in the registry under Software\Interactive Intelligence\Eic\Notifier\InteractionIDSequencer. The system updates this value for every 10000 allocations.

The Call ID Key encoding that is stored in the database also changed. The Call ID Key, for version 2.1, is still a ten character case-sensitive string.

The Client Call ID Key: 1JKBx00001

It is unique over a long period of time (roughly 127 years), but its construction is now largely opaque due to the need to handle a much larger range of identifiers than was previously necessary. The Call ID Key is constructed based on the following information:

- Date of the last server restart
- Site ID
- Six least significant digits of the Call ID (*when the six least significant digits of the Call ID overflow, the portion of the Call ID Key derived from the server restart date is advanced*.)

How Client Call IDs were generated in 2.0 and prior

IC stores Client Call IDs in the call attribute, *EIC_CallId*, which serves as a temporary unique identifier for each call object. The decimal value of the Client Call ID is an integer in the range 100000000 – 4199999999. The call identifier is typically a large number composed of 10 digits in the following format:

MM SSS DDDDD

DDDDD is a number that the system increments for each successive Client Call ID. It is set to zero when then system clock rolls over at midnight, and is set to the number of seconds since midnight when IC is started.

SSS is the site identifier. To maintain unique identifiers across multi-site installations, IC uses the value of the Site Identifier (an integer value in the range 0 - 999 configured by IA) in the Client Call ID.

MM is, by default, the day of the month. If your IC system consumes Client Call IDs at a very high rate and the DDDDD value overflows, IC resets DDDDD to zero and MM to (MM + 17) % 41 + 1.

Because IC uses this method the system reuses Client Call IDs approximately every month.

For example, 0100100001 is the first Client Call ID the system allocates at site 1 on the first day of any month (the first Client Call ID is not necessarily the first call made or received).

As another example, 1200200100 is the hundredth Client Call ID the system allocates at site 2 on the twelfth day of any month.

Description of EIC_CallIDKey in 2.0 and Prior

In order to have unique keys for database operations, we added a new call attribute, EIC_CallIdKey.

EIC_CallIDKey is a unique identifier whose value is a ten-character string composed of alphanumeric characters (digits and upper and lower case alphabetic) that is unique across related sites over an extended period of time (106 years). The first three characters of Eic_CallIDKey correspond to a rare base 62 count of the number of days since we released IC. The next two characters are the Site ID (it looks like decimal digits for Site IDs in the range 0 - 99 and looks like strange alphabetic characters for Site IDs larger than 99). The last five characters correspond to the low order five digits of the original IC 1.2 Call ID.

Typically, there is a one-to-one mapping between the Client Call ID value and the Eic_CallIDKey string. When IC supports intelligent transfers between related IC systems, the system keeps the same Client Call ID and Eic_CallIDKey value with the call.

Appendix B: Setting the QueuePeriodStatisticsInterval Server Parameter

The installation sets the Queue Period Statistics (QPS) interval in Interaction Administrator using the server parameter, QueuePeriodStatisticsInterval. The default value of the parameter is 1800 seconds (30 minutes) at the time of installation, but you can change it. You can set the value to anything greater than 300 seconds. There is a very high upper limit (max value = max signed integer, in case you were wondering), but the practical range for most sites is from 300 to 3600 seconds. The default IC reports generally expect a one-hour or less interval setting but the reports will work with any setting. If you set an interval that is not an even part of an hour it leads to odd-looking statistics, but that is not necessarily wrong, it just may be hard for you to read the statistics. You can set a one-day interval setting, but this is unlikely to give you meaningful statistics on a busy IC server.

If you change this parameter it will affect the number of records stored in the SQL tables, therefore, it will affect the size of the database.

If you change this parameter it will not affect records IC has already stored in the database. If you run any QPS reports using a date/time range that includes a date/time in which a different setting was in effect, the report will reflect the setting in effect when the records were written to the database.

The QPS process of writing statistics to the SQL tables may cause additional processing load on your server. If you change the parameter it will affect how often the QPS processing load occurs and you should considered this if you have a heavily-used server that is near its maximum processor capacity.

In some cases, the interval start times might not be exactly on the hour. Stopping and starting IC usually causes this. You might see these odd interval start times in the reports.

If you change the value of the parameter, you do not have to reboot the server for this parameter change to take effect. IC processes the change dynamically and it takes effect immediately. The system reschedules the current end of the interval with the new value.

Although it often appears that way, the setting is not necessarily always relative to the hour. The setting is scheduled from midnight. Therefore, as long as 60 is divisible by the value without a remainder, the hour is one of the times. However, nothing restricts it to this behavior.

If you change the setting to run every 5 minutes, then QPS will run at the next 5-minute period. For example, if you change the setting to 5 minutes at 12:01, then QPS will run at 12:05. If you change it to one hour at 12:30 PM, then it will run at 1:00 PM.

Change Log

The following table summarizes the changes made to the *Reporting Data Dictionary* technical reference document version 2.4, since product availability.

Change Log Table

Change	**Date**
Updated current copyright information	July 6, 2006
Reporting available in Interaction Supervisor	July 7, 2006
Updated cType Hierarchy example values for the Agent Queue Statistics Interval log and the Workgroup Queue Statistics Interval table	September 28, 2006

Index